

# Mobile-C

## 1.10.9

Generated by Doxygen 1.6.1

Wed Mar 17 15:31:55 2010



# Contents



# Chapter 1

# MobileC

## 1.1 Introduction

Welcome to the MobileC documentation. For a quick user-api reference, please refer to the file [libmc.h](#).

This documentation is provided supplementary to the main Mobile-C User Guide. The official user guide may be obtained at <http://www.mobilec.org> , and should also be included with your Mobile-C Distribution package in the file docs/mobilec.pdf .



## **Chapter 2**

# **Installing LibMC.NET**

Installing LibMC.NET is straightforward but involves several steps.

## 2.1 Requirements

In order to use LibMC.NET you will need the following:

- Ch version 6.0.0 or greater from <http://www.softintegration.com/>.
- Embedded Ch version 6.0.0 or greater, also from <http://www.softintegration.com/>.
- Mobile-C 1.10.0 or greater. See Section [Downloading Mobile-C](#) for instructions on how to obtain Mobile-C.
- Visual Studio 2005 or later. Express versions of Visual Studio can be found at <http://www.microsoft.com/express/>.

## 2.2 Downloading Mobile-C

First, you must obtain a version of the Mobile-C source code. If you are reading this, chances are you have already completed this step. If you have not already downloaded the source code, it can be done in one of three ways:

- Download a supported release of Mobile-C. Visit [the Mobile-C website](#) for more information on supported releases.
- Download the latest source (unsupported) from [Sourceforge](#). This will give you the most current version of Mobile-C, but not necessarily the most stable version.
- Check out the latest source code from the SVN repository. This requires that you have a subversion client installed. More information can be found at [this location](#).

## 2.3 Building the Mobile-C Libraries

Once you have obtained the Mobile-C source, please see the Mobile-C User's Guide for information on compiling Mobile-C under Windows. Currently, only the Visual Studio .NET 2005 project is supported for LibMC.NET. Section 2.3 of the User's Guide describes how to compile Mobile-C into a static library. For LibMC.NET, at least one of two configurations are required: the "Debug\_DLL" or "Release\_DLL" versions. To build either one, select the appropriate configuration (this replaces step 3 in the User's Guide, Section 2.3.1) and build the solution (step 4). Alternatively, you may select "Batch Build" from the "Build" menu and build all four possible configurations.

## 2.4 Install the Mobile-C Libraries

After building the Mobile-C DLL files, the project will automatically copy the files to the system directory. By default, the files are copied to C:/Windows/System32/. If your system is configured differently or you wish to change the installation directory, right-click on the mc\_lib\_win32 project in the Solution Explorer and select "Properties." In the mc\_lib\_win32 Property Pages treeview, select "Configuration Properties," then "Build Events," and finally "Post-Build Event." You can then change the "Command Line" field to copy the files to the directory of your choice. If you change the installation directory, be sure that your



chosen directory is in the system path and that you remove any other versions of the files. You will also need to execute a "Rebuild" on the project to ensure the files are copied to the new location.

## 2.5 Build LibMC.NET

Once you have built the Mobile-C DLL files, you can build LibMC.NET. Open the LibMC.NET solution file located in the directory you installed or checked out Mobile-C to at src/win32/LibMC.NET/LibMC.Net.sln. From the "Build" menu, select "Rebuild Solution." You may want to build both the "Debug" and "Release" versions, or perform a batch build as described previously.



## **Chapter 3**

# **Getting Started**

LibMC.NET is very easy to use. The demo programs provided with the download are a good place to start. Please see the Examples section for more information.

### 3.1 Build the Demo Programs

The LibMC.NET demo program solution is located in the directory you installed or checked out Mobile-C to at `demos/win32/LibMC.NET/LibMCDemos.sln`. As before, select "Rebuild Solution" from the "Build" menu. Note that the demo program solution contains the LibMC.NET project as well. You may also build LibMC.NET from within the demo program solution.

By default, the LibMCGui demo is selected in the demo program solution. You may run this program by selecting the "Debug" menu then "Start Debugging" or by pressing F5. Other demo programs can be started by right-clicking the project in the Solution Explorer and selecting "Debug" then "Start new instance." The demo programs have their own documentation as well. See the README file in `src/win32/LibMC.NET` or `demos/win32/LibMC.NET` for information on how to build the demo program documentation.

## **Chapter 4**

# **Using LibMC.NET**

This section explains how to use the LibMC.NET class library in your .NET project. Currently, it only describes the process for using the library in a C# console or GUI application. Other languages, such as VB and managed C++, will require similar actions.

## 4.1 Create a Project

First, create the type of project you would like to use from the Visual Studio "Start Page" or the "File" menu. Select the name and location of the project as you would any other project. Second, add a reference to the configuration of LibMC.NET you would like to use. For debugging purposes, the "Debug" configuration is probably best. To add the reference, right click the "References" item in the Solution Explorer for the project you just created. Select the "Browse" tab and navigate to the output directory of the LibMC.NET project. The directory is located at `src/win32/LibMC.NET/bin/Configuration/` in the Mobile-C source directory, where Configuration is either "Debug" or "Release." Select the DLL file and click "Ok." The References item in the Solution Explorer should now list "LibMC." Be sure to save the solution at this point.

## 4.2 Using LibMC.NET Classes and Functions

As with any other namespace, you must add the declaration "using LibMC;" to any file you want to have access to the class libraries. Once you have added the using statement, you can declare objects from the library as you normally would declare any other objects. See the example programs for more details.

## 4.3 Other Options

You may want to enable one or more features in your project that can help you use LibMC.NET or debug problems. If you add any XML files to your project, you probably will want to set their properties in the project to copy the files to the output directory. This is done by selecting the file in the Solution Explorer, opening its properties, and setting two fields:

- Set the "Build Action" field to "Content" if it is not already set. This will make the file part of the project should you decide to publish or package it.
- Set the "Copy to output directory" to "Copy if newer" or "Copy always." This will copy the file when you build the project.

There is also one important note regarding XML files in Visual Studio. *Do not create XML files from within Visual Studio.* The Visual Studio XML file template contains a few leading characters that specify the encoding of the file. They are hidden and you will not be able to change them. These characters are not currently supported by Mobile-C and will crash a receiving agency.

To open the project properties, right-click the project in the Solution Explorer and select "Properties." In the "Debug" pane, you may wish to set an alternate working directory for the project if you want easy access to XML files outside of the project. This is useful for debugging, but may result in errors finding files if you package the project or create an installer. In general, it is best to specify all files with full paths because the Mobile-C library loads from a different location than the project. If you would like to be able to debug the Mobile-C library, you should select the "Enable unmanaged code debugging" check box. This will allow you to more easily see any errors that may occur in the unmanaged library, though hopefully none will.

## **Chapter 5**

# **Common Operations**

This section contains examples of commonly used operations for three main [LibMC](#) classes:

- [MCAgency](#) The mobile agent agency.
- [MCAgent](#) Mobile agents.
- [MCAclMessage](#) Agent communication language messages.

For complete programs and more detailed examples, see the Examples section.

## 5.1 MCAgency

Examples of commonly used MCAgency operations:

Declare an agency as a member of a class:

```
public static MCAgency Agency = new MCAgency();
```

Set the agency's port:

```
int temp = 5051;
Agency.Port = temp;
```

Start an agency:

```
int temp;
temp = Agency.Initialize();
if (temp != 0)
    Console.WriteLine("Initialize: " + temp.ToString());
```

Pause and resume an agency:

```
Agency.HaltAgency();
Agency.ResumeAgency();
```

Turn off the command prompt thread:

```
temp = Agency.SetThreadOff(MCAgency.MC_ThreadIndex_e.MC_THREAD_CP);
if (temp != 0)
    Console.WriteLine("SetThreadOff: " + temp.ToString());
```

Load an agent into a local agency:

```
String filename = "agent.xml";
try
{
    Agency.LoadAgentMigrationMessageFile(filename);
}
catch (Exception ex)
{
    Console.WriteLine("Error loading file: " + ex.Message);
}
```

Ideally, the file name should be specified absolutely.

Send an agent to a remote agency:



```
String filename = "agent.xml";
String ip = "192.168.23.93";
int port = 5051;
try
{
    Agency.SendAgentMigrationMessageFile(filename, ip, port);
}
catch (Exception ex)
{
    Console.WriteLine("Error sending file: " + ex.Message);
}
```

Find an agent by name:

```
MCAgent agent;
try
{
    agent = Agency.FindAgentByName("persistent1");
}
catch (Exception e)
{
    Console.WriteLine("Exception: " + e.Message);
}
```

Wait for an agent to arrive:

```
MCAgent agent;
Agency.ResetSignal();
try
{
    agent = Agency.WaitRetrieveAgent();
}
catch (Exception e)
{
    Console.WriteLine("Exception: " + e.Message);
}
```

Wait indefinitely while an agency runs:

```
Agency.MainLoop();
```

## 5.2 MCAgent

Examples of commonly used MCAgent operations: Find an agent by name (assumes an MCAgency named Agency):

```
MCAgent agent;
try
{
    agent = Agency.FindAgentByName("persistent1");
}
catch (Exception e)
{
    Console.WriteLine("Exception: " + e.Message);
}
```

Terminate an agent

```
int temp;
try
{

```

```

        temp = agent.TerminateAgent();
        Console.WriteLine("TerminateAgent() returned " +
            temp.ToString() + ".");
    }
    catch (Exception e)
    {
        Console.WriteLine("Exception: " + e.Message);
    }

```

Print information about an agent:

```

Console.WriteLine(agent.ToString());
Console.WriteLine(agent.GetAgentXMLString());
Console.WriteLine(agent.RetrieveAgentCode());

```

## 5.3 MCAclMessage

Examples of commonly used MCAclMessage operations:

Create a new, blank ACL message:

```

MCAclMessage tmp = new MCAclMessage();
tmp.New();

```

Set the performative field:

```

tmp.SetPerformative(MCAclMessage.MC_FipaPerformative_e.FIPA_INFORM);

```

Set the sender:

```

tmp.SetSender("agency", "http://" + host + ":" +
    localport.ToString() + "/acc");

```

Add an alternate reply-to field:

```

tmp.AddReplyTo("mobagent2", "http://" + host + ":" +
    localport.ToString() + "/acc");

```

Add a receiver to the message:

```

tmp.AddReceiver("mobagent1", "http://" + host + ":" +
    localport.ToString() + "/acc");

```

Set the content of the message:

```

tmp.SetContent("This is content. Yay!");

```

Finally, send and destroy the message:

```

Agency.AclSend(tmp);
tmp.Destroy();

```

Note that messages contain a pointer to allocated unmanaged memory and need to be disposed of after they are used. The agency creates a copy of the message when it is sent, and therefore the MCAclMessage object is no longer needed.

## **Chapter 6**

### **Todo List**

Global **LibMC::MCAgency::RegisterService**(MCAgent agent, int agentID, String agentName, String[] serviceNames, i  
Test MC\_RegisterService and MC\_SearchForService.

Global **LibMC::MCAgency::SearchForService**(String searchString, IntPtr agentNames, IntPtr serviceNames, IntPtr ag  
Implement SearchForService

Global **LibMC::MCAgency::Steer**(IntPtr funcptr, IntPtr arg) Implement MC\_Steer

Global **LibMC::MCAgency::SteerControl**() Test MC\_SteerControl, MC\_Steer.

Global **LibMC::MCAgent::GetAgentExecEngine**() Wrap MC\_GetAgentExecEngine with an object  
for the void\* pointer return type (Ch interpreter).

Global **LibMC::MCAgent::GetAgentReturnData**(int task\_num, IntPtr data, IntPtr dim, IntPtr extent)  
Implement GetAgentReturnData

## **Chapter 7**

### **Bug List**

Global **LibMC::MCAgency::SearchForService**(String searchString, IntPtr agentNames, IntPtr serviceNames, IntPtr arg) MC\_SearchForService is not yet implemented.

Global **LibMC::MCAgency::Steer**(IntPtr funcptr, IntPtr arg) MC\_Steer is not yet implemented.

# Chapter 8

## Namespace Index

### 8.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">EmbeddedCh</a>	.....	??
<a href="#">LibMC</a>	(Namespace for the .NET wrapper for Mobile-C ) .....	??
<a href="#">LibMC::Properties</a>	(Namespace for the .NET wrapper properties class ) .....	??
<a href="#">Program1</a>	.....	??





# Chapter 9

## Data Structure Index

### 9.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">_hr_time</a>	??
<a href="#">_ssl_context</a>	??
<a href="#">_ssl_session</a>	??
<a href="#">_x509_buf</a>	??
<a href="#">_x509_cert</a>	??
<a href="#">_x509_name</a>	??
<a href="#">_x509_node</a>	??
<a href="#">_x509_raw</a>	??
<a href="#">_x509_time</a>	??
<a href="#">aes_context</a> (AES context structure )	??
<a href="#">agency_s</a> (The agency handle )	??
<a href="#">agent_datastate_s</a>	??
<a href="#">agent_mailbox_s</a>	??
<a href="#">agent_s</a>	??
<a href="#">agent_task_s</a>	??
<a href="#">agent_thread_arg_s</a>	??
<a href="#">AP_GENERIC_s</a>	??
<a href="#">arc4_context</a> (ARC4 context structure )	??
<a href="#">barrier_node_s</a>	??
<a href="#">barrier_queue_s</a>	??
<a href="#">EmbeddedCh::ChBlock_t</a>	??
<a href="#">EmbeddedCh::ChInfo_t</a>	??
<a href="#">EmbeddedCh::ChInterp</a>	??
<a href="#">EmbeddedCh::ChMemInfo_t</a>	??
<a href="#">EmbeddedCh::ChOptions_t</a>	??
<a href="#">LibMC::MCAgency::ChOptions_t</a> (ChOptions structures )	??
<a href="#">EmbeddedCh::ChUserDefinedInfo_t</a>	??
<a href="#">EmbeddedCh::ChUserDefinedTag</a>	??
<a href="#">EmbeddedCh::ChVaList</a>	??
<a href="#">cmd_prompt_s</a>	??
<a href="#">command_s</a>	??
<a href="#">connection_s</a>	??
<a href="#">fipa_expression_s::content_u</a>	??

<a href="#">des3_context</a> (Triple-DES context structure ) . . . . .	??
<a href="#">des_context</a> (DES context structure ) . . . . .	??
<a href="#">dhm_context</a> . . . . .	??
<a href="#">dynstring_s</a> . . . . .	??
<a href="#">fipa_acl_envelope_Received_s</a> . . . . .	??
<a href="#">fipa_acl_envelope_s</a> . . . . .	??
<a href="#">fipa_acl_message_s</a> . . . . .	??
<a href="#">fipa_acl_Param_s</a> . . . . .	??
<a href="#">fipa_agent_identifier_s</a> . . . . .	??
<a href="#">fipa_agent_identifier_set_s</a> . . . . .	??
<a href="#">fipa_comm_action_t</a> . . . . .	??
<a href="#">fipa_comm_message_check_t</a> . . . . .	??
<a href="#">fipa_comm_performative_t</a> . . . . .	??
<a href="#">fipa_comm_protocol_cn_t</a> . . . . .	??
<a href="#">fipa_comm_protocol_t</a> . . . . .	??
<a href="#">fipa_comm_reply_t</a> . . . . .	??
<a href="#">fipa_comm_t</a> . . . . .	??
<a href="#">fipa_DateTime_s</a> . . . . .	??
<a href="#">fipa_expression_s</a> . . . . .	??
<a href="#">fipa_list_t</a> . . . . .	??
<a href="#">fipa_message_string_s</a> . . . . .	??
<a href="#">fipa_number_s</a> . . . . .	??
<a href="#">fipa_string_s</a> . . . . .	??
<a href="#">fipa_url_s</a> . . . . .	??
<a href="#">fipa_url_sequence_s</a> . . . . .	??
<a href="#">fipa_word_s</a> . . . . .	??
<a href="#">foo_c</a> . . . . .	??
<a href="#">foo_s</a> . . . . .	??
<a href="#">havege_state</a> (HAVEGE state structure ) . . . . .	??
<a href="#">host_id_s</a> . . . . .	??
<a href="#">hr_time</a> (Timer structure ) . . . . .	??
<a href="#">interpreter_variable_data_s</a> . . . . .	??
<a href="#">LibMC::InvalidAgencyException</a> (Exception class for use with null agency pointers ) . . . . .	??
<a href="#">LibMC::InvalidAgentException</a> (Exception class for use with null agent pointers ) . . . . .	??
<a href="#">list_s</a> . . . . .	??
<a href="#">listNode_s</a> . . . . .	??
<a href="#">mc_platform_s</a> . . . . .	??
<a href="#">mc_rwlock_s</a> . . . . .	??
<a href="#">LibMC::MCAclMessage</a> (Encapsulates ACL messages in the Mobile-C library ) . . . . .	??
<a href="#">LibMC::MCAgency</a> (Wrapper class for <a href="#">MCAgency_t</a> structure ) . . . . .	??
<a href="#">LibMC::MCAgency::MCAgency_t</a> . . . . .	??
<a href="#">MCAgencyOptions_s</a> (User modifiable agency <a href="#">options</a> ) . . . . .	??
<a href="#">LibMC::MCAgency::MCAgencyOptions_t</a> . . . . .	??
<a href="#">LibMC::MCAgent</a> (Wrapper class for <a href="#">MCAgent_t</a> structure ) . . . . .	??
<a href="#">md2_context</a> (MD2 context structure ) . . . . .	??
<a href="#">md4_context</a> (MD4 context structure ) . . . . .	??
<a href="#">md5_context</a> (MD5 context structure ) . . . . .	??
<a href="#">message_s</a> . . . . .	??
<a href="#">message_send_arg_s</a> . . . . .	??
<a href="#">mpi</a> (MPI structure ) . . . . .	??
<a href="#">mtp_http_content_s</a> . . . . .	??
<a href="#">mtp_http_s</a> . . . . .	??
<a href="#">mxml_attr_s</a> . . . . .	??
<a href="#">mxml_custom_s</a> . . . . .	??

<a href="#">mxml_fdbuf_s</a>	??
<a href="#">mxml_index_s</a>	??
<a href="#">mxml_node_s</a>	??
<a href="#">mxml_text_s</a>	??
<a href="#">mxml_value_s</a>	??
<a href="#">mxml_value_u</a>	??
<a href="#">options</a>	??
<a href="#">Program1::Program</a>	??
<a href="#">rsa_context</a> (RSA context structure )	??
<a href="#">LibMC::Properties::Settings</a>	??
<a href="#">sha1_context</a> (SHA-1 context structure )	??
<a href="#">sha2_context</a> (SHA-256 context structure )	??
<a href="#">sha4_context</a> (SHA-512 context structure )	??
<a href="#">syncList_s</a>	??
<a href="#">syncListNode_s</a>	??



# Chapter 10

## File Index

### 10.1 File List

Here is a list of all files with brief descriptions:

/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/acc.c	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/agent.c	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/agent_datastate.c	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/agent_mailbox.c	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/agent_return_data.c	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/agent_task.c	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/ams.c	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/barrier.c	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/cmd_prompt.c	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/connection.c	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/data_structures.c	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/df.c	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/dynstring.c	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/fipa_acl.c	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/fipa_envelope.c	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/libmc.c	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mc_platform.c	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mc_rwlock.c	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/message.c	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/message_queue.c	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mtp_http.c	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/winconfig.h	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/xml_compose.c	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/xml_helper.c	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/xml_parser.c	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/acc.h	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/agent.h	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/agent_datastate.h	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/agent_lib.h	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/agent_mailbox.h	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/agent_task.h	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/ams.h	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/ap_queue_template.h	??

/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/barrier.h	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/cmd_prompt.h	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/commands.h	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/commands.x.h	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/connection.h	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/data_structures.h	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/df.h	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/df_request.x.h	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/dynstring.h	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/fipa_acl.h	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/fipa_acl_envelope.h	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/fipa_comm.h	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/host_id.h	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/interpreter_variable_data.h	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/libmc.h (MobileC api header file)	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/macros.h	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/mc_error.h	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/mc_platform.h	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/mc_rwlock.h	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/message.h	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/mobilec.h	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/mtp_http.h	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/xml_compose.h	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/xml_helper.h	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/xml_parser.h	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mc_list/list.c	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mc_list/list.h	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mc_sync/sync_list.c	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mc_sync/sync_list.h	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mxml-2.2.2/config.h	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mxml-2.2.2/mxml-attr.c	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mxml-2.2.2/mxml-entity.c	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mxml-2.2.2/mxml-file.c	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mxml-2.2.2/mxml-index.c	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mxml-2.2.2/mxml-node.c	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mxml-2.2.2/mxml-private.c	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mxml-2.2.2/mxml-search.c	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mxml-2.2.2/mxml-set.c	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mxml-2.2.2/mxml-string.c	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mxml-2.2.2/mxml.h	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mxml-2.2.2/mxmldoc.c	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mxml-2.2.2/testmxml.c	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mxml-2.2.2/test/class.cxx	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mxml-2.2.2/test/enum.cxx	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mxml-2.2.2/test/function.cxx	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mxml-2.2.2/test/struct.cxx	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mxml-2.2.2/vcnet/config.h	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mxml-2.2.2/vcnet2005/config.h	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mxml-2.2.2/vcnet2008/config.h	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/asm.c	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/asm.h	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/asm_message_composer.c	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/asm_message_composer.h	??

/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/asm_message_parser.c . . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/asm_message_parser.h . . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/asm_node.c . . . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/asm_node.h . . . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/interface.c . . . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/interface.h . . . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/mc_dh.c . . . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/mc_dh.h . . . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/aes.h .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/arc4.h .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/base64.h . . . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/bignum.h . . . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/bn_-mul.h . . . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/certs.h	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/config.h . . . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/debug.h . . . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/des.h .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/dhm.h .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/havege.h . . . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/md2.h .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/md4.h .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/md5.h .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/net.h .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/openssl.h . . . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/padlock.h . . . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/rsa.h .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/sha1.h	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/sha2.h	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/sha4.h	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/ssl.h .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/timing.h . . . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/x509.h	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/library/aes.c . . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/library/arc4.c . . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/library/base64.c . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/library/bignum.c . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/library/certs.c . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/library/debug.c . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/library/des.c . . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/library/dhm.c . . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/library/havege.c . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/library/md2.c . . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/library/md4.c . . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/library/md5.c . . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/library/net.c . . . .	??

/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/library/padlock.c . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/library/rsa.c . . . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/library/sha1.c . . . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/library/sha2.c . . . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/library/sha4.c . . . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/library/ssl_cli.c . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/library/ssl_srv.c . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/library/ssl_tls.c . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/library/timing.c . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/library/x509parse.c . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/programs/aes/aesencrypt2.c . . . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/programs/hash/hello.c . . . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/programs/hash/md5sum.c . . . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/programs/hash/sha1sum.c . . . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/programs/hash/sha2sum.c . . . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/programs/pkey/dh_client.c . . . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/programs/pkey/dh_genprime.c . . . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/programs/pkey/dh_server.c . . . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/programs/pkey/mpi_demo.c . . . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/programs/pkey/rsa_genkey.c . . . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/programs/pkey/rsa_sign.c . . . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/programs/pkey/rsa_verify.c . . . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/programs/ssl/ssl_client1.c . . . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/programs/ssl/ssl_client2.c . . . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/programs/ssl/ssl_server.c . . . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/programs/test/benchmark.c . . . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/programs/test/selftest.c . . . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/programs/test/ssl_test.c . . . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/util/mc_genkey.c . . . . .	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/win32/EmbeddedCh.Net/EmbeddedCh.Net/Ch.cs	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/win32/EmbeddedCh.Net/EmbeddedCh.Net/ChInterp.cs	??
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/win32/EmbeddedCh.Net/EmbeddedCh.Net/ChUserDefinedTag.cs	??



/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/win32/EmbeddedCh.Net/EmbeddedCh.Net/[ChVaList.cs](#)  
??  
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/win32/EmbeddedCh.Net/EmbeddedCh.Net/[EmbedCh.cs](#)  
??  
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/win32/EmbeddedCh.Net/EmbeddedCh.Net/Properties/[AssemblyInfo.cs](#)  
??  
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/win32/EmbeddedCh.Net/Program1/[Program.cs](#)  
??  
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/win32/EmbeddedCh.Net/Program1/Properties/[AssemblyInfo.cs](#)  
??  
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/win32/LibMC.net/LibMC/[MCAclMessage.cs](#)  
??  
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/win32/LibMC.net/LibMC/[MCAgency.cs](#)  
??  
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/win32/LibMC.net/LibMC/[MCAgent.cs](#) ??  
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/win32/LibMC.net/LibMC/[MCExports.cs](#)  
??  
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/win32/LibMC.net/LibMC/[Settings.cs](#) . ??  
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/win32/LibMC.net/LibMC/Properties/[AssemblyInfo.cs](#)  
??  
/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/win32/LibMC.net/LibMC/Properties/[Settings.Designer.cs](#)  
??



# Chapter 11

## Namespace Documentation

### 11.1 EmbeddedCh Namespace Reference

#### Data Structures

- struct [ChInfo\\_t](#)
- class [ChInterp](#)
- class [ChUserDefinedTag](#)
- class [ChVaList](#)
- struct [ChOptions\\_t](#)
- struct [ChBlock\\_t](#)
- struct [ChUserDefinedInfo\\_t](#)
- struct [ChMemInfo\\_t](#)

#### Enumerations

- enum [ChType\\_t](#) {  
    [CH\\_UNDEFINETYPE](#), [CH\\_CHARTYPE](#) = 10, [CH\\_UCHARTYPE](#), [CH\\_SHORTTYPE](#),  
    [CH\\_USHORTTYPE](#), [CH\\_INTTYPE](#), [CH\\_UINTTYPE](#), [CH\\_LLINTTYPE](#),  
    [CH\\_ULLINTTYPE](#), [CH\\_FLOATTYPE](#), [CH\\_DOUBLETTYPE](#), [CH\\_LDOUBLETTYPE](#),  
    [CH\\_COMPLEXTYPE](#), [CH\\_LCOMPLEXTYPE](#), [CH\\_STRINGTYPE](#), [CH\\_FILETYPE](#),  
    [CH\\_VOIDTYPE](#), [CH\\_PROCTYPE](#), [CH\\_STRUCTTYPE](#), [CH\\_CLASSTYPE](#),  
    [CH\\_UNIONTYPE](#), [CH\\_ENUMTYPE](#), [CH\\_CARRAYTYPE](#) = 80, [CH\\_CARRAYPTRTYPE](#),  
    [CH\\_CARRAYVLATTYPE](#), [CH\\_CHARRAYTYPE](#), [CH\\_CHARRAYPTRTYPE](#), [CH\\_-](#)  
    [CHARRAYVLATTYPE](#),  
    [CH\\_NULLTYPE](#) = 100, [CH\\_VOIDPTRTYPE](#), [CH\\_CHARPTRTYPE](#), [CH\\_UCHARPTRTYPE](#),  
    [CH\\_SHORTPTRTYPE](#), [CH\\_USHORTPTRTYPE](#), [CH\\_INTPTRTYPE](#), [CH\\_UINTPTRTYPE](#),  
    [CH\\_LLINTPTRTYPE](#), [CH\\_ULLINTPTRTYPE](#), [CH\\_FLOATPTRTYPE](#), [CH\\_DOUBLEPTRTYPE](#),  
    [CH\\_LDOUBLEPTRTYPE](#), [CH\\_COMPLEXPTRTYPE](#), [CH\\_LCOMPLEXPTRTYPE](#), [CH\\_-](#)  
    [STRINGPTRTYPE](#),  
    [CH\\_PROCPTRTYPE](#), [CH\\_FILEPTRTYPE](#), [CH\\_STRUCTPTRTYPE](#), [CH\\_CLASSPTRTYPE](#),  
    [CH\\_UNIONPTRTYPE](#), [CH\\_ENUMPTRTYPE](#), [CH\\_VOIDPTR2TYPE](#) = 200, [CH\\_-](#)  
    [CHARPTR2TYPE](#),  
}

```

CH_UCHARPTR2TYPE,    CH_SHORTPTR2TYPE,    CH_USHORTPTR2TYPE,    CH_-
INTPTR2TYPE,
CH_UINTPTR2TYPE,    CH_LLINTPTR2TYPE,    CH_ULLINTPTR2TYPE,    CH_-
FLOATPTR2TYPE,
CH_DOUBLEPTR2TYPE,    CH_LDOUBLEPTR2TYPE,    CH_COMPLEXPTR2TYPE,    CH_-
LCOMPLEXPTR2TYPE,
CH_STRINGPTR2TYPE,    CH_FILEPTR2TYPE,    CH_STRUCTPTR2TYPE,    CH_-
CLASSPTR2TYPE,
CH_UNIONPTR2TYPE, CH_ENUMPTR2TYPE }
• enum ChRetVal { CH_OK = 0, CH_ERROR = -1, CH_ABORT = 1 }
• enum ChFuncType_t {
    CH_NOTFUNCTYPE, CH_FUNCTYPE, CH_FUNCPROTOTYPE, CH_FUNCPTRTYPE,
    CH_FUNCMEMBERTYPE, CH_FUNCCONSTYPE, CH_FUNCDESTTYPE }
• enum ChVarType_t { CH_NOTVARTYPE, CH_GLOBALVARTYPE, CH_LOCALVARTYPE }
• enum ChShellType { CH_REGULARCH = 0, CH_SAFECH = 1 }
    Ch shell type.

• enum ChFileDescriptor { STDIN_FILENO = 0, STDOUT_FILENO = 1, STDERR_FILENO = 2 }
• enum ChCallbackMask {
    CH_MASKNONE = 0X0000, CH_MASKCALL = 0X0001, CH_MASKRET = 0X0002, CH_-
    MASKBLOCK = 0X0004,
    CH_MASKEND = 0X0008, CH_MASKLINE = 0X0010, CH_MASKCOUNT = 0X0020, CH_-
    MASKABORT = 0X0040 }

```

## 11.1.1 Enumeration Type Documentation

### 11.1.1.1 enum EmbeddedCh::ChCallbackMask

Enumerator:

```

CH_MASKNONE
CH_MASKCALL
CH_MASKRET
CH_MASKBLOCK
CH_MASKEND
CH_MASKLINE
CH_MASKCOUNT
CH_MASKABORT

```

Definition at line 126 of file EmbedCh.cs.

### 11.1.1.2 enum EmbeddedCh::ChFileDescriptor

Enumerator:

```

STDIN_FILENO
STDOUT_FILENO
STDERR_FILENO

```

Definition at line 117 of file EmbedCh.cs.

#### 11.1.1.3 enum EmbeddedCh::ChFuncType\_t

Enumerator:

*CH\_NOTFUNCTYPE*  
*CH\_FUNCTYPE*  
*CH\_FUNCPROTOTYPE*  
*CH\_FUNCPTRTYPE*  
*CH\_FUNCMEMBERTYPE*  
*CH\_FUNCCONSTTYPE*  
*CH\_FUNCDESTTYPE*

Definition at line 24 of file EmbedCh.cs.

#### 11.1.1.4 enum EmbeddedCh::ChRetVal

Enumerator:

*CH\_OK*  
*CH\_ERROR*  
*CH\_ABORT*

Definition at line 106 of file Ch.cs.

#### 11.1.1.5 enum EmbeddedCh::ChShellType

Ch shell type. Used to set the shell type for the Ch interpreter.

Enumerator:

*CH\_REGULARCH* Default, regular shell  
*CH\_SAFECH* Safe shell

Definition at line 110 of file EmbedCh.cs.

#### 11.1.1.6 enum EmbeddedCh::ChType\_t

Enumerator:

*CH\_UNDEFINETYPE*  
*CH\_CHARTYPE*  
*CH\_UCHARTYPE*  
*CH\_SHORTTYPE*  
*CH\_USHORTTYPE*  
*CH\_INTTYPE*  
*CH\_UINTTYPE*  
*CH\_LLINTTYPE*  
*CH\_ULLINTTYPE*

*CH\_FLOATTYPE*  
*CH\_DOUBLETTYPE*  
*CH\_LDOUBLETTYPE*  
*CH\_COMPLEXTYPE*  
*CH\_LCOMPLEXTYPE*  
*CH\_STRINGTYPE*  
*CH\_FILETYPE*  
*CH\_VOIDTYPE*  
*CH\_PROCTYPE*  
*CH\_STRUCTTYPE*  
*CH\_CLASSTYPE*  
*CH\_UNIONTYPE*  
*CH\_ENUMTYPE*  
*CH\_CARRAYTYPE*  
*CH\_CARRAYPTRTYPE*  
*CH\_CARRAYVLATTYPE*  
*CH\_CHARRAYTYPE*  
*CH\_CHARRAYPTRTYPE*  
*CH\_CHARRAYVLATTYPE*  
*CH\_NULLTYPE*  
*CH\_VOIDPTRTYPE*  
*CH\_CHARPTRTYPE*  
*CH\_UCHARPTRTYPE*  
*CH\_SHORTPTRTYPE*  
*CH\_USHORTPTRTYPE*  
*CH\_INTPTRTYPE*  
*CH\_UINTPTRTYPE*  
*CH\_LLINTPTRTYPE*  
*CH\_ULLINTPTRTYPE*  
*CH\_FLOATPTRTYPE*  
*CH\_DOUBLEPTRTYPE*  
*CH\_LDOUBLEPTRTYPE*  
*CH\_COMPLEXPTRTYPE*  
*CH\_LCOMPLEXPTRTYPE*  
*CH\_STRINGPTRTYPE*  
*CH\_PROCPTRTYPE*  
*CH\_FILEPTRTYPE*  
*CH\_STRUCTPTRTYPE*  
*CH\_CLASSPTRTYPE*  
*CH\_UNIONPTRTYPE*  
*CH\_ENUMPTRTYPE*

*CH\_VOIDPTR2TYPE*  
*CH\_CHARPTR2TYPE*  
*CH\_UCHARPTR2TYPE*  
*CH\_SHORTPTR2TYPE*  
*CH\_USHORTPTR2TYPE*  
*CH\_INTPTR2TYPE*  
*CH\_UINTPTR2TYPE*  
*CH\_LLINTPTR2TYPE*  
*CH\_ULLINTPTR2TYPE*  
*CH\_FLOATPTR2TYPE*  
*CH\_DOUBLEPTR2TYPE*  
*CH\_LDOUBLEPTR2TYPE*  
*CH\_COMPLEXPTR2TYPE*  
*CH\_LCOMPLEXPTR2TYPE*  
*CH\_STRINGPTR2TYPE*  
*CH\_FILEPTR2TYPE*  
*CH\_STRUCTPTR2TYPE*  
*CH\_CLASSPTR2TYPE*  
*CH\_UNIONPTR2TYPE*  
*CH\_ENUMPTR2TYPE*

Definition at line 19 of file Ch.cs.

#### 11.1.1.7 enum EmbeddedCh::ChVarType\_t

**Enumerator:**

*CH\_NOTVARTYPE*  
*CH\_GLOBALVARTYPE*  
*CH\_LOCALVARTYPE*

Definition at line 36 of file EmbedCh.cs.

## 11.2 LibMC Namespace Reference

Namespace for the .NET wrapper for Mobile-C.

### Namespaces

- namespace [Properties](#)  
*Namespace for the .NET wrapper properties class.*

### Data Structures

- class [MCAclMessage](#)  
*Encapsulates ACL messages in the Mobile-C library.*
- class [MCAgency](#)  
*Wrapper class for [MCAgency\\_t](#) structure.*
- class [InvalidAgencyException](#)  
*Exception class for use with null agency pointers.*
- class [MCAgent](#)  
*Wrapper class for [MCAgent\\_t](#) structure.*
- class [InvalidAgentException](#)  
*Exception class for use with null agent pointers.*

### 11.2.1 Detailed Description

Namespace for the .NET wrapper for Mobile-C. [LibMC](#) encapsulates the Mobile-C DLL for windows in an .NET class library. .NET programs can access the library to create agencies, connect to agencies, interact with agents, etc.



## 11.3 LibMC::Properties Namespace Reference

Namespace for the .NET wrapper properties class.

### Data Structures

- class [Settings](#)

#### 11.3.1 Detailed Description

Namespace for the .NET wrapper properties class. Any user or global properties that should be preserved from session to session can be added here through the designer. There are currently no properties in use.

## 11.4 Program1 Namespace Reference

### Data Structures

- class [Program](#)

## Chapter 12

# Data Structure Documentation

### 12.1 `_hr_time` Struct Reference

#### Data Fields

- struct timeval [start](#)

#### 12.1.1 Detailed Description

Definition at line 45 of file `timing.c`.

#### 12.1.2 Field Documentation

##### 12.1.2.1 `struct timeval _hr_time::start` [`read`]

Definition at line 47 of file `timing.c`.

Referenced by `get_timer()`.

The documentation for this struct was generated from the following file:

- `/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/library/timing.c`

## 12.2 `_ssl_context` Struct Reference

```
#include <ssl.h>
```

### Data Fields

- `int` `state`
- `int` `major_ver`
- `int` `minor_ver`
- `int` `max_major_ver`
- `int` `max_minor_ver`
- `int(* f_rng)(void *)`
- `void(* f_dbg)(void *, int, char *)`
- `int(* f_recv)(void *, unsigned char *, int)`
- `int(* f_send)(void *, unsigned char *, int)`
- `void *` `p_rng`
- `void *` `p_dbg`
- `void *` `p_recv`
- `void *` `p_send`
- `int` `resume`
- `int` `timeout`
- `ssl_session *` `session`
- `int(* s_get)(ssl_context *)`
- `int(* s_set)(ssl_context *)`
- `unsigned char *` `in_ctr`
- `unsigned char *` `in_hdr`
- `unsigned char *` `in_msg`
- `unsigned char *` `in_offt`
- `int` `in_msgtype`
- `int` `in_msglen`
- `int` `in_left`
- `int` `in_hhlen`
- `int` `nb_zero`
- `unsigned char *` `out_ctr`
- `unsigned char *` `out_hdr`
- `unsigned char *` `out_msg`
- `int` `out_msgtype`
- `int` `out_msglen`
- `int` `out_left`
- `rsa_context *` `rsa_key`
- `x509_cert *` `own_cert`
- `x509_cert *` `ca_chain`
- `x509_cert *` `peer_cert`
- `char *` `peer_cn`
- `int` `endpoint`
- `int` `authmode`
- `int` `client_auth`
- `int` `verify_result`
- `dhm_context` `dhm_ctx`

- `md5_context` `fin_md5`
- `sha1_context` `fin_sha1`
- `int` `do_crypt`
- `int *` `ciphers`
- `int` `pmslen`
- `int` `keylen`
- `int` `minlen`
- `int` `ivlen`
- `int` `maclen`
- `unsigned char` `randbytes` [64]
- `unsigned char` `premaster` [256]
- `unsigned char` `iv_enc` [16]
- `unsigned char` `iv_dec` [16]
- `unsigned char` `mac_enc` [32]
- `unsigned char` `mac_dec` [32]
- `unsigned long` `ctx_enc` [128]
- `unsigned long` `ctx_dec` [128]
- `unsigned char *` `hostname`
- `unsigned long` `hostname_len`

### 12.2.1 Detailed Description

Definition at line 149 of file `ssl.h`.

### 12.2.2 Field Documentation

#### 12.2.2.1 `int _ssl_context::authmode`

verification mode

Definition at line 220 of file `ssl.h`.

Referenced by `ssl_parse_certificate()`, `ssl_set_authmode()`, and `ssl_write_certificate_request()`.

#### 12.2.2.2 `x509_cert* _ssl_context::ca_chain`

own trusted CA chain

Definition at line 215 of file `ssl.h`.

Referenced by `ssl_parse_certificate()`, `ssl_set_ca_chain()`, and `ssl_write_certificate_request()`.

#### 12.2.2.3 `int* _ssl_context::ciphers`

allowed ciphersuites

Definition at line 232 of file `ssl.h`.

Referenced by `ssl_parse_client_hello()`, `ssl_parse_server_hello()`, `ssl_set_ciphers()`, and `ssl_write_client_hello()`.

**12.2.2.4 int \_ssl\_context::client\_auth**

flag for client auth.

Definition at line 221 of file ssl.h.

Referenced by ssl\_parse\_certificate\_request(), ssl\_parse\_server\_hello\_done(), ssl\_write\_certificate(), and ssl\_write\_certificate\_verify().

**12.2.2.5 unsigned long \_ssl\_context::ctx\_dec[128]**

decryption context

Definition at line 249 of file ssl.h.

Referenced by ssl\_decrypt\_buf(), and ssl\_derive\_keys().

**12.2.2.6 unsigned long \_ssl\_context::ctx\_enc[128]**

encryption context

Definition at line 248 of file ssl.h.

Referenced by ssl\_derive\_keys(), and ssl\_encrypt\_buf().

**12.2.2.7 dhm\_context \_ssl\_context::dhm\_ctx**

DHM key exchange

Definition at line 227 of file ssl.h.

Referenced by ssl\_free(), ssl\_parse\_client\_key\_exchange(), ssl\_parse\_server\_key\_exchange(), ssl\_set\_dh\_param(), ssl\_write\_client\_key\_exchange(), and ssl\_write\_server\_key\_exchange().

**12.2.2.8 int \_ssl\_context::do\_crypt**

en(de)cryption flag

Definition at line 231 of file ssl.h.

Referenced by ssl\_parse\_change\_cipher\_spec(), ssl\_parse\_finished(), ssl\_read\_record(), ssl\_write\_change\_cipher\_spec(), ssl\_write\_finished(), and ssl\_write\_record().

**12.2.2.9 int \_ssl\_context::endpoint**

0: client, 1: server

Definition at line 219 of file ssl.h.

Referenced by ssl\_derive\_keys(), ssl\_handshake(), ssl\_parse\_certificate(), ssl\_parse\_finished(), ssl\_set\_endpoint(), ssl\_write\_certificate(), and ssl\_write\_finished().

**12.2.2.10 void(\* \_ssl\_context::f\_dbg)(void \*,int, char \*)**

Referenced by debug\_print\_buf(), debug\_print\_crt(), debug\_print\_mpi(), debug\_print\_msg(), debug\_print\_ret(), and ssl\_set\_dbg().

**12.2.2.11 `int(* _ssl_context::f_recv)(void *, unsigned char *, int)`**

Referenced by `ssl_fetch_input()`, and `ssl_set_bio()`.

**12.2.2.12 `int(* _ssl_context::f_rng)(void *)`**

Referenced by `ssl_parse_client_key_exchange()`, `ssl_set_rng()`, `ssl_write_client_hello()`, `ssl_write_client_key_exchange()`, `ssl_write_server_hello()`, and `ssl_write_server_key_exchange()`.

**12.2.2.13 `int(* _ssl_context::f_send)(void *, unsigned char *, int)`**

Referenced by `ssl_flush_output()`, and `ssl_set_bio()`.

**12.2.2.14 `md5_context _ssl_context::fin_md5`**

Finished MD5 checksum

Definition at line 228 of file `ssl.h`.

Referenced by `ssl_calc_verify()`, `ssl_init()`, `ssl_parse_client_hello()`, `ssl_parse_finished()`, `ssl_read_record()`, `ssl_write_finished()`, and `ssl_write_record()`.

**12.2.2.15 `sha1_context _ssl_context::fin_sha1`**

Finished SHA-1 checksum

Definition at line 229 of file `ssl.h`.

Referenced by `ssl_calc_verify()`, `ssl_init()`, `ssl_parse_client_hello()`, `ssl_parse_finished()`, `ssl_read_record()`, `ssl_write_finished()`, and `ssl_write_record()`.

**12.2.2.16 `unsigned char* _ssl_context::hostname`**

Definition at line 254 of file `ssl.h`.

Referenced by `ssl_free()`, `ssl_init()`, `ssl_set_hostname()`, and `ssl_write_client_hello()`.

**12.2.2.17 `unsigned long _ssl_context::hostname_len`**

Definition at line 255 of file `ssl.h`.

Referenced by `ssl_free()`, `ssl_init()`, `ssl_set_hostname()`, and `ssl_write_client_hello()`.

**12.2.2.18 `unsigned char* _ssl_context::in_ctr`**

64-bit incoming message counter

Definition at line 187 of file `ssl.h`.

Referenced by `ssl_decrypt_buf()`, `ssl_free()`, and `ssl_init()`.

**12.2.2.19 unsigned char\* \_ssl\_context::in\_hdr**

5-byte record header (in\_ctr+8)

Definition at line 188 of file ssl.h.

Referenced by ssl\_decrypt\_buf(), ssl\_fetch\_input(), ssl\_init(), ssl\_parse\_client\_hello(), and ssl\_read\_record().

**12.2.2.20 int \_ssl\_context::in\_hslen**

current handshake message length

Definition at line 196 of file ssl.h.

Referenced by ssl\_parse\_certificate(), ssl\_parse\_certificate\_verify(), ssl\_parse\_client\_key\_exchange(), ssl\_parse\_finished(), ssl\_parse\_server\_hello(), ssl\_parse\_server\_hello\_done(), ssl\_parse\_server\_key\_exchange(), and ssl\_read\_record().

**12.2.2.21 int \_ssl\_context::in\_left**

amount of data read so far

Definition at line 194 of file ssl.h.

Referenced by ssl\_fetch\_input(), ssl\_parse\_client\_hello(), and ssl\_read\_record().

**12.2.2.22 unsigned char\* \_ssl\_context::in\_msg**

the message contents (in\_hdr+5)

Definition at line 189 of file ssl.h.

Referenced by ssl\_decrypt\_buf(), ssl\_init(), ssl\_parse\_certificate(), ssl\_parse\_certificate\_request(), ssl\_parse\_certificate\_verify(), ssl\_parse\_change\_cipher\_spec(), ssl\_parse\_client\_hello(), ssl\_parse\_client\_key\_exchange(), ssl\_parse\_finished(), ssl\_parse\_server\_hello(), ssl\_parse\_server\_hello\_done(), ssl\_parse\_server\_key\_exchange(), ssl\_read(), and ssl\_read\_record().

**12.2.2.23 int \_ssl\_context::in\_msglen**

record header: message length

Definition at line 193 of file ssl.h.

Referenced by ssl\_decrypt\_buf(), ssl\_get\_bytes\_avail(), ssl\_parse\_certificate(), ssl\_parse\_change\_cipher\_spec(), ssl\_read(), and ssl\_read\_record().

**12.2.2.24 int \_ssl\_context::in\_msgtype**

record header: message type

Definition at line 192 of file ssl.h.

Referenced by ssl\_decrypt\_buf(), ssl\_parse\_certificate(), ssl\_parse\_certificate\_request(), ssl\_parse\_certificate\_verify(), ssl\_parse\_change\_cipher\_spec(), ssl\_parse\_client\_key\_exchange(), ssl\_parse\_



`finished()`, `ssl_parse_server_hello()`, `ssl_parse_server_hello_done()`, `ssl_parse_server_key_exchange()`, `ssl_read()`, and `ssl_read_record()`.

#### 12.2.2.25 `unsigned char* _ssl_context::in_offt`

read offset in application data

Definition at line 190 of file `ssl.h`.

Referenced by `ssl_get_bytes_avail()`, and `ssl_read()`.

#### 12.2.2.26 `unsigned char _ssl_context::iv_dec[16]`

IV (decryption)

Definition at line 243 of file `ssl.h`.

Referenced by `ssl_decrypt_buf()`, and `ssl_derive_keys()`.

#### 12.2.2.27 `unsigned char _ssl_context::iv_enc[16]`

IV (encryption)

Definition at line 242 of file `ssl.h`.

Referenced by `ssl_derive_keys()`, and `ssl_encrypt_buf()`.

#### 12.2.2.28 `int _ssl_context::ivlen`

IV length

Definition at line 236 of file `ssl.h`.

Referenced by `ssl_decrypt_buf()`, `ssl_derive_keys()`, and `ssl_encrypt_buf()`.

#### 12.2.2.29 `int _ssl_context::keylen`

symmetric key length

Definition at line 234 of file `ssl.h`.

Referenced by `ssl_derive_keys()`.

#### 12.2.2.30 `unsigned char _ssl_context::mac_dec[32]`

MAC (decryption)

Definition at line 246 of file `ssl.h`.

Referenced by `ssl_decrypt_buf()`, and `ssl_derive_keys()`.

#### 12.2.2.31 `unsigned char _ssl_context::mac_enc[32]`

MAC (encryption)

Definition at line 245 of file ssl.h.

Referenced by `ssl_derive_keys()`, and `ssl_encrypt_buf()`.

#### **12.2.2.32 int \_ssl\_context::maclen**

MAC length

Definition at line 237 of file ssl.h.

Referenced by `ssl_decrypt_buf()`, `ssl_derive_keys()`, and `ssl_encrypt_buf()`.

#### **12.2.2.33 int \_ssl\_context::major\_ver**

equal to `SSL_MAJOR_VERSION_3`

Definition at line 156 of file ssl.h.

Referenced by `ssl_parse_client_hello()`, `ssl_read_record()`, `ssl_write_client_hello()`, `ssl_write_record()`, and `ssl_write_server_hello()`.

#### **12.2.2.34 int \_ssl\_context::max\_major\_ver**

max. major version from client

Definition at line 159 of file ssl.h.

Referenced by `ssl_parse_client_hello()`, `ssl_parse_client_key_exchange()`, `ssl_write_client_hello()`, and `ssl_write_client_key_exchange()`.

#### **12.2.2.35 int \_ssl\_context::max\_minor\_ver**

max. minor version from client

Definition at line 160 of file ssl.h.

Referenced by `ssl_parse_client_hello()`, `ssl_parse_client_key_exchange()`, `ssl_write_client_hello()`, and `ssl_write_client_key_exchange()`.

#### **12.2.2.36 int \_ssl\_context::minlen**

min. ciphertext length

Definition at line 235 of file ssl.h.

Referenced by `ssl_decrypt_buf()`, `ssl_derive_keys()`, and `ssl_read_record()`.

#### **12.2.2.37 int \_ssl\_context::minor\_ver**

either 0 (SSL3) or 1 (TLS1.0)

Definition at line 157 of file ssl.h.

Referenced by `ssl_calc_finished()`, `ssl_calc_verify()`, `ssl_decrypt_buf()`, `ssl_derive_keys()`, `ssl_encrypt_buf()`, `ssl_parse_certificate()`, `ssl_parse_client_hello()`, `ssl_parse_client_key_exchange()`, `ssl_parse_`

`finished()`, `ssl_parse_server_hello()`, `ssl_read_record()`, `ssl_write_certificate()`, `ssl_write_client_hello()`, `ssl_write_client_key_exchange()`, `ssl_write_finished()`, `ssl_write_record()`, and `ssl_write_server_hello()`.

#### 12.2.2.38 `int _ssl_context::nb_zero`

# of 0-length encrypted messages

Definition at line 197 of file `ssl.h`.

Referenced by `ssl_decrypt_buf()`.

#### 12.2.2.39 `unsigned char* _ssl_context::out_ctr`

64-bit outgoing message counter

Definition at line 202 of file `ssl.h`.

Referenced by `ssl_encrypt_buf()`, `ssl_free()`, and `ssl_init()`.

#### 12.2.2.40 `unsigned char* _ssl_context::out_hdr`

5-byte record header (`out_ctr+8`)

Definition at line 203 of file `ssl.h`.

Referenced by `ssl_flush_output()`, `ssl_init()`, and `ssl_write_record()`.

#### 12.2.2.41 `int _ssl_context::out_left`

amount of data not yet written

Definition at line 208 of file `ssl.h`.

Referenced by `ssl_flush_output()`, `ssl_write()`, and `ssl_write_record()`.

#### 12.2.2.42 `unsigned char* _ssl_context::out_msg`

the message contents (`out_hdr+5`)

Definition at line 204 of file `ssl.h`.

Referenced by `ssl_close_notify()`, `ssl_encrypt_buf()`, `ssl_init()`, `ssl_write()`, `ssl_write_certificate()`, `ssl_write_certificate_request()`, `ssl_write_certificate_verify()`, `ssl_write_change_cipher_spec()`, `ssl_write_client_hello()`, `ssl_write_client_key_exchange()`, `ssl_write_finished()`, `ssl_write_record()`, `ssl_write_server_hello()`, `ssl_write_server_hello_done()`, and `ssl_write_server_key_exchange()`.

#### 12.2.2.43 `int _ssl_context::out_msglen`

record header: message length

Definition at line 207 of file `ssl.h`.

Referenced by `ssl_close_notify()`, `ssl_encrypt_buf()`, `ssl_flush_output()`, `ssl_write()`, `ssl_write_certificate()`, `ssl_write_certificate_request()`, `ssl_write_certificate_verify()`, `ssl_write_change_cipher_spec()`, `ssl_write_client_hello()`, `ssl_write_client_key_exchange()`, `ssl_write_finished()`, `ssl_write_record()`, `ssl_write_server_hello()`, `ssl_write_server_hello_done()`, and `ssl_write_server_key_exchange()`.

**12.2.2.44 int \_ssl\_context::out\_msgtype**

record header: message type

Definition at line 206 of file ssl.h.

Referenced by ssl\_close\_notify(), ssl\_encrypt\_buf(), ssl\_write(), ssl\_write\_certificate(), ssl\_write\_certificate\_request(), ssl\_write\_certificate\_verify(), ssl\_write\_change\_cipher\_spec(), ssl\_write\_client\_hello(), ssl\_write\_client\_key\_exchange(), ssl\_write\_finished(), ssl\_write\_record(), ssl\_write\_server\_hello(), ssl\_write\_server\_hello\_done(), and ssl\_write\_server\_key\_exchange().

**12.2.2.45 x509\_cert\* \_ssl\_context::own\_cert**

own X.509 certificate

Definition at line 214 of file ssl.h.

Referenced by ssl\_set\_own\_cert(), and ssl\_write\_certificate().

**12.2.2.46 void\* \_ssl\_context::p\_dbg**

context for the debug function

Definition at line 171 of file ssl.h.

Referenced by debug\_print\_buf(), debug\_print\_crt(), debug\_print\_mpi(), debug\_print\_msg(), debug\_print\_ret(), and ssl\_set\_dbg().

**12.2.2.47 void\* \_ssl\_context::p\_recv**

context for reading operations

Definition at line 172 of file ssl.h.

Referenced by ssl\_fetch\_input(), and ssl\_set\_bio().

**12.2.2.48 void\* \_ssl\_context::p\_rng**

context for the RNG function

Definition at line 170 of file ssl.h.

Referenced by ssl\_parse\_client\_key\_exchange(), ssl\_set\_rng(), ssl\_write\_client\_hello(), ssl\_write\_client\_key\_exchange(), ssl\_write\_server\_hello(), and ssl\_write\_server\_key\_exchange().

**12.2.2.49 void\* \_ssl\_context::p\_send**

context for writing operations

Definition at line 173 of file ssl.h.

Referenced by ssl\_flush\_output(), and ssl\_set\_bio().

**12.2.2.50 x509\_cert\* \_ssl\_context::peer\_cert**

peer X.509 cert chain

Definition at line 216 of file `ssl.h`.

Referenced by `main()`, `ssl_free()`, `ssl_parse_certificate()`, `ssl_parse_certificate_verify()`, `ssl_parse_server_key_exchange()`, and `ssl_write_client_key_exchange()`.

#### 12.2.2.51 `char* _ssl_context::peer_cn`

expected peer CN

Definition at line 217 of file `ssl.h`.

Referenced by `ssl_parse_certificate()`, and `ssl_set_ca_chain()`.

#### 12.2.2.52 `int _ssl_context::pmslen`

premaster length

Definition at line 233 of file `ssl.h`.

Referenced by `ssl_derive_keys()`, `ssl_parse_client_key_exchange()`, and `ssl_write_client_key_exchange()`.

#### 12.2.2.53 `unsigned char _ssl_context::premaster[256]`

premaster secret

Definition at line 240 of file `ssl.h`.

Referenced by `ssl_derive_keys()`, `ssl_parse_client_key_exchange()`, and `ssl_write_client_key_exchange()`.

#### 12.2.2.54 `unsigned char _ssl_context::randbytes[64]`

random bytes

Definition at line 239 of file `ssl.h`.

Referenced by `ssl_derive_keys()`, `ssl_parse_client_hello()`, `ssl_parse_server_hello()`, `ssl_parse_server_key_exchange()`, `ssl_write_client_hello()`, `ssl_write_server_hello()`, and `ssl_write_server_key_exchange()`.

#### 12.2.2.55 `int _ssl_context::resume`

session resuming flag

Definition at line 178 of file `ssl.h`.

Referenced by `my_get_session()`, `ssl_derive_keys()`, `ssl_parse_finished()`, `ssl_parse_server_hello()`, `ssl_set_session()`, `ssl_write_client_hello()`, `ssl_write_finished()`, and `ssl_write_server_hello()`.

#### 12.2.2.56 `rsa_context* _ssl_context::rsa_key`

own RSA private key

Definition at line 213 of file `ssl.h`.

Referenced by `ssl_parse_client_key_exchange()`, `ssl_set_own_cert()`, `ssl_write_certificate_verify()`, and `ssl_write_server_key_exchange()`.

**12.2.2.57 int(\* \_ssl\_context::s\_get)(ssl\_context \*)**

(server) get callback

Referenced by `ssl_set_scb()`, and `ssl_write_server_hello()`.

**12.2.2.58 int(\* \_ssl\_context::s\_set)(ssl\_context \*)**

(server) set callback

Referenced by `ssl_parse_client_key_exchange()`, and `ssl_set_scb()`.

**12.2.2.59 ssl\_session\* \_ssl\_context::session**

current session data

Definition at line 180 of file `ssl.h`.

Referenced by `my_get_session()`, `my_set_session()`, `ssl_calc_finished()`, `ssl_calc_verify()`, `ssl_derive_keys()`, `ssl_get_cipher()`, `ssl_parse_client_hello()`, `ssl_parse_client_key_exchange()`, `ssl_parse_server_hello()`, `ssl_parse_server_key_exchange()`, `ssl_set_session()`, `ssl_write_client_hello()`, `ssl_write_client_key_exchange()`, `ssl_write_server_hello()`, and `ssl_write_server_key_exchange()`.

**12.2.2.60 int \_ssl\_context::state**

SSL handshake: current state

Definition at line 154 of file `ssl.h`.

Referenced by `ssl_close_notify()`, `ssl_handshake_client()`, `ssl_handshake_server()`, `ssl_parse_certificate()`, `ssl_parse_certificate_request()`, `ssl_parse_certificate_verify()`, `ssl_parse_change_cipher_spec()`, `ssl_parse_client_hello()`, `ssl_parse_client_key_exchange()`, `ssl_parse_finished()`, `ssl_parse_server_hello()`, `ssl_parse_server_hello_done()`, `ssl_parse_server_key_exchange()`, `ssl_read()`, `ssl_write()`, `ssl_write_certificate()`, `ssl_write_certificate_request()`, `ssl_write_certificate_verify()`, `ssl_write_change_cipher_spec()`, `ssl_write_client_hello()`, `ssl_write_client_key_exchange()`, `ssl_write_finished()`, `ssl_write_server_hello()`, `ssl_write_server_hello_done()`, and `ssl_write_server_key_exchange()`.

**12.2.2.61 int \_ssl\_context::timeout**

sess. expiration time

Definition at line 179 of file `ssl.h`.

Referenced by `my_get_session()`, `my_set_session()`, `ssl_set_session()`, and `ssl_write_client_hello()`.

**12.2.2.62 int \_ssl\_context::verify\_result**

verification result

Definition at line 222 of file `ssl.h`.

Referenced by `ssl_get_verify_result()`, and `ssl_parse_certificate()`.

The documentation for this struct was generated from the following file:

- `/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/ssl.h`

## 12.3 \_ssl\_session Struct Reference

```
#include <ssl.h>
```

### Data Fields

- [time\\_t start](#)
- [int cipher](#)
- [int length](#)
- unsigned char [id](#) [32]
- unsigned char [master](#) [48]
- [ssl\\_session](#) \* [next](#)

### 12.3.1 Detailed Description

Definition at line 139 of file `ssl.h`.

### 12.3.2 Field Documentation

#### 12.3.2.1 `int _ssl_session::cipher`

chosen cipher

Definition at line 142 of file `ssl.h`.

Referenced by `my_get_session()`, `ssl_derive_keys()`, `ssl_get_cipher()`, `ssl_parse_client_hello()`, `ssl_parse_client_key_exchange()`, `ssl_parse_server_hello()`, `ssl_parse_server_key_exchange()`, `ssl_write_client_key_exchange()`, `ssl_write_server_hello()`, and `ssl_write_server_key_exchange()`.

#### 12.3.2.2 `unsigned char _ssl_session::id[32]`

session identifier

Definition at line 144 of file `ssl.h`.

Referenced by `my_get_session()`, `my_set_session()`, `ssl_parse_client_hello()`, `ssl_parse_server_hello()`, `ssl_write_client_hello()`, and `ssl_write_server_hello()`.

#### 12.3.2.3 `int _ssl_session::length`

session id length

Definition at line 143 of file `ssl.h`.

Referenced by `my_get_session()`, `my_set_session()`, `ssl_parse_client_hello()`, `ssl_parse_server_hello()`, `ssl_write_client_hello()`, and `ssl_write_server_hello()`.

#### 12.3.2.4 `unsigned char _ssl_session::master[48]`

the master secret

Definition at line 145 of file `ssl.h`.

Referenced by `my_get_session()`, `ssl_calc_finished()`, `ssl_calc_verify()`, and `ssl_derive_keys()`.

#### **12.3.2.5 `ssl_session* _ssl_session::next`**

next session entry

Definition at line 146 of file `ssl.h`.

Referenced by `main()`, `my_get_session()`, and `my_set_session()`.

#### **12.3.2.6 `time_t _ssl_session::start`**

starting time

Definition at line 141 of file `ssl.h`.

Referenced by `my_get_session()`, `my_set_session()`, `ssl_parse_server_hello()`, and `ssl_write_client_hello()`.

The documentation for this struct was generated from the following file:

- `/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/ssl.h`



## 12.4 \_x509\_buf Struct Reference

```
#include <x509.h>
```

### Data Fields

- [int tag](#)
- [int len](#)
- unsigned char \* [p](#)

### 12.4.1 Detailed Description

Definition at line 96 of file x509.h.

### 12.4.2 Field Documentation

#### 12.4.2.1 int \_x509\_buf::len

Definition at line 99 of file x509.h.

Referenced by `ssl_write_certificate()`, `ssl_write_certificate_request()`, `x509_free()`, `x509_get_alg()`, `x509_get_ext()`, `x509_get_name()`, `x509_get_pubkey()`, `x509_get_serial()`, `x509_get_sig()`, `x509_get_uid()`, `x509parse_cert_info()`, `x509parse_crt()`, `x509parse_dn_gets()`, and `x509parse_verify()`.

#### 12.4.2.2 unsigned char\* \_x509\_buf::p

Definition at line 100 of file x509.h.

Referenced by `ssl_write_certificate()`, `ssl_write_certificate_request()`, `x509_free()`, `x509_get_alg()`, `x509_get_ext()`, `x509_get_name()`, `x509_get_pubkey()`, `x509_get_serial()`, `x509_get_sig()`, `x509_get_uid()`, `x509parse_cert_info()`, `x509parse_crt()`, `x509parse_dn_gets()`, and `x509parse_verify()`.

#### 12.4.2.3 int \_x509\_buf::tag

Definition at line 98 of file x509.h.

Referenced by `x509_get_alg()`, `x509_get_ext()`, `x509_get_name()`, `x509_get_serial()`, `x509_get_sig()`, and `x509_get_uid()`.

The documentation for this struct was generated from the following file:

- [/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/x509.h](#)

## 12.5 `_x509_cert` Struct Reference

```
#include <x509.h>
```

### Data Fields

- `x509_buf raw`
- `x509_buf tbs`
- `int version`
- `x509_buf serial`
- `x509_buf sig_oid1`
- `x509_buf issuer_raw`
- `x509_buf subject_raw`
- `x509_name issuer`
- `x509_name subject`
- `x509_time valid_from`
- `x509_time valid_to`
- `x509_buf pk_oid`
- `rsa_context rsa`
- `x509_buf issuer_id`
- `x509_buf subject_id`
- `x509_buf v3_ext`
- `int ca_istrue`
- `int max_pathlen`
- `x509_buf sig_oid2`
- `x509_buf sig`
- `struct _x509_cert * next`

### 12.5.1 Detailed Description

Definition at line 119 of file `x509.h`.

### 12.5.2 Field Documentation

#### 12.5.2.1 `int _x509_cert::ca_istrue`

Definition at line 144 of file `x509.h`.

Referenced by `x509parse_cert()`, and `x509parse_verify()`.

#### 12.5.2.2 `x509_name _x509_cert::issuer`

Definition at line 131 of file `x509.h`.

Referenced by `x509_free()`, `x509parse_cert_info()`, and `x509parse_cert()`.

#### 12.5.2.3 `x509_buf _x509_cert::issuer_id`

Definition at line 140 of file `x509.h`.

Referenced by `x509parse_cert()`.

#### 12.5.2.4 `x509_buf_x509_cert::issuer_raw`

Definition at line 128 of file `x509.h`.

Referenced by `x509parse_cert()`, and `x509parse_verify()`.

#### 12.5.2.5 `int_x509_cert::max_pathlen`

Definition at line 145 of file `x509.h`.

Referenced by `x509parse_cert()`, and `x509parse_verify()`.

#### 12.5.2.6 `struct_x509_cert*_x509_cert::next` [read]

Definition at line 150 of file `x509.h`.

Referenced by `debug_print_cert()`, `main()`, `ssl_test()`, `ssl_write_certificate()`, `ssl_write_certificate_request()`, `x509_free()`, `x509parse_cert()`, and `x509parse_verify()`.

#### 12.5.2.7 `x509_buf_x509_cert::pk_oid`

Definition at line 137 of file `x509.h`.

Referenced by `x509parse_cert()`.

#### 12.5.2.8 `x509_buf_x509_cert::raw`

Definition at line 121 of file `x509.h`.

Referenced by `ssl_write_certificate()`, `x509_free()`, and `x509parse_cert()`.

#### 12.5.2.9 `rsa_context_x509_cert::rsa`

Definition at line 138 of file `x509.h`.

Referenced by `debug_print_cert()`, `ssl_parse_certificate_verify()`, `ssl_parse_server_key_exchange()`, `ssl_write_client_key_exchange()`, `x509_free()`, `x509parse_cert_info()`, `x509parse_cert()`, and `x509parse_verify()`.

#### 12.5.2.10 `x509_buf_x509_cert::serial`

Definition at line 125 of file `x509.h`.

Referenced by `x509parse_cert_info()`, and `x509parse_cert()`.

#### 12.5.2.11 `x509_buf_x509_cert::sig`

Definition at line 148 of file `x509.h`.

Referenced by `x509parse_cert()`, and `x509parse_verify()`.

**12.5.2.12 x509\_buf\_x509\_cert::sig\_oid1**

Definition at line 126 of file x509.h.

Referenced by x509parse\_cert\_info(), x509parse\_crt(), and x509parse\_verify().

**12.5.2.13 x509\_buf\_x509\_cert::sig\_oid2**

Definition at line 147 of file x509.h.

Referenced by x509parse\_crt().

**12.5.2.14 x509\_name\_x509\_cert::subject**

Definition at line 132 of file x509.h.

Referenced by x509\_free(), x509parse\_cert\_info(), x509parse\_crt(), and x509parse\_verify().

**12.5.2.15 x509\_buf\_x509\_cert::subject\_id**

Definition at line 141 of file x509.h.

Referenced by x509parse\_crt().

**12.5.2.16 x509\_buf\_x509\_cert::subject\_raw**

Definition at line 129 of file x509.h.

Referenced by ssl\_write\_certificate\_request(), x509parse\_crt(), and x509parse\_verify().

**12.5.2.17 x509\_buf\_x509\_cert::tbs**

Definition at line 122 of file x509.h.

Referenced by x509parse\_crt(), and x509parse\_verify().

**12.5.2.18 x509\_buf\_x509\_cert::v3\_ext**

Definition at line 142 of file x509.h.

Referenced by x509parse\_crt().

**12.5.2.19 x509\_time\_x509\_cert::valid\_from**

Definition at line 134 of file x509.h.

Referenced by x509parse\_cert\_info(), and x509parse\_crt().

**12.5.2.20 x509\_time\_x509\_cert::valid\_to**

Definition at line 135 of file x509.h.

Referenced by x509parse\_cert\_info(), x509parse\_crt(), and x509parse\_expired().

**12.5.2.21 `int _x509_cert::version`**

Definition at line 124 of file `x509.h`.

Referenced by `x509parse_cert_info()`, `x509parse_cert()`, and `x509parse_verify()`.

The documentation for this struct was generated from the following file:

- `/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/x509.h`

## 12.6 `_x509_name` Struct Reference

```
#include <x509.h>
```

### Data Fields

- [x509\\_buf oid](#)
- [x509\\_buf val](#)
- [struct `\_x509\_name` \\* next](#)

### 12.6.1 Detailed Description

Definition at line 104 of file `x509.h`.

### 12.6.2 Field Documentation

#### 12.6.2.1 `struct _x509_name* _x509_name::next` **[read]**

Definition at line 108 of file `x509.h`.

Referenced by `x509_free()`, `x509_get_name()`, `x509parse_dn_gets()`, and `x509parse_verify()`.

#### 12.6.2.2 `x509_buf _x509_name::oid`

Definition at line 106 of file `x509.h`.

Referenced by `x509_get_name()`, `x509parse_dn_gets()`, and `x509parse_verify()`.

#### 12.6.2.3 `x509_buf _x509_name::val`

Definition at line 107 of file `x509.h`.

Referenced by `x509_get_name()`, `x509parse_dn_gets()`, and `x509parse_verify()`.

The documentation for this struct was generated from the following file:

- [/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/x509.h](#)

## 12.7 `_x509_node` Struct Reference

```
#include <x509.h>
```

### Data Fields

- unsigned char \* [data](#)
- unsigned char \* [p](#)
- unsigned char \* [end](#)
- size\_t [len](#)

### 12.7.1 Detailed Description

Definition at line 157 of file `x509.h`.

### 12.7.2 Field Documentation

#### 12.7.2.1 unsigned char\* `_x509_node::data`

Definition at line 159 of file `x509.h`.

#### 12.7.2.2 unsigned char\* `_x509_node::end`

Definition at line 161 of file `x509.h`.

#### 12.7.2.3 size\_t `_x509_node::len`

Definition at line 163 of file `x509.h`.

#### 12.7.2.4 unsigned char\* `_x509_node::p`

Definition at line 160 of file `x509.h`.

The documentation for this struct was generated from the following file:

- `/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/x509.h`

## 12.8 `_x509_raw` Struct Reference

```
#include <x509.h>
```

### Data Fields

- [x509\\_node raw](#)
- [x509\\_node tbs](#)
- [x509\\_node version](#)
- [x509\\_node serial](#)
- [x509\\_node tbs\\_signalg](#)
- [x509\\_node issuer](#)
- [x509\\_node validity](#)
- [x509\\_node subject](#)
- [x509\\_node subpubkey](#)
- [x509\\_node signalg](#)
- [x509\\_node sign](#)

### 12.8.1 Detailed Description

Definition at line 167 of file x509.h.

### 12.8.2 Field Documentation

#### 12.8.2.1 `x509_node _x509_raw::issuer`

Definition at line 175 of file x509.h.

#### 12.8.2.2 `x509_node _x509_raw::raw`

Definition at line 169 of file x509.h.

#### 12.8.2.3 `x509_node _x509_raw::serial`

Definition at line 173 of file x509.h.

#### 12.8.2.4 `x509_node _x509_raw::sign`

Definition at line 181 of file x509.h.

#### 12.8.2.5 `x509_node _x509_raw::signalg`

Definition at line 180 of file x509.h.

#### 12.8.2.6 `x509_node _x509_raw::subject`

Definition at line 177 of file x509.h.



**12.8.2.7 `x509_node_x509_raw::subpubkey`**

Definition at line 178 of file `x509.h`.

**12.8.2.8 `x509_node_x509_raw::tbs`**

Definition at line 170 of file `x509.h`.

**12.8.2.9 `x509_node_x509_raw::tbs_signalg`**

Definition at line 174 of file `x509.h`.

**12.8.2.10 `x509_node_x509_raw::validity`**

Definition at line 176 of file `x509.h`.

**12.8.2.11 `x509_node_x509_raw::version`**

Definition at line 172 of file `x509.h`.

The documentation for this struct was generated from the following file:

- `/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/x509.h`

## 12.9 `_x509_time` Struct Reference

```
#include <x509.h>
```

### Data Fields

- `int year`
- `int mon`
- `int day`
- `int hour`
- `int min`
- `int sec`

#### 12.9.1 Detailed Description

Definition at line 112 of file `x509.h`.

#### 12.9.2 Field Documentation

##### 12.9.2.1 `int _x509_time::day`

Definition at line 114 of file `x509.h`.

Referenced by `x509_get_dates()`, `x509parse_cert_info()`, and `x509parse_expired()`.

##### 12.9.2.2 `int _x509_time::hour`

Definition at line 115 of file `x509.h`.

Referenced by `x509_get_dates()`, and `x509parse_cert_info()`.

##### 12.9.2.3 `int _x509_time::min`

Definition at line 115 of file `x509.h`.

Referenced by `x509_get_dates()`, and `x509parse_cert_info()`.

##### 12.9.2.4 `int _x509_time::mon`

Definition at line 114 of file `x509.h`.

Referenced by `x509_get_dates()`, `x509parse_cert_info()`, and `x509parse_expired()`.

##### 12.9.2.5 `int _x509_time::sec`

Definition at line 115 of file `x509.h`.

Referenced by `x509_get_dates()`, and `x509parse_cert_info()`.

### 12.9.2.6 `int _x509_time::year`

Definition at line 114 of file `x509.h`.

Referenced by `x509_get_dates()`, `x509parse_cert_info()`, and `x509parse_expired()`.

The documentation for this struct was generated from the following file:

- `/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/x509.h`

## 12.10 aes\_context Struct Reference

AES context structure.

```
#include <aes.h>
```

### Data Fields

- `int nr`
- `unsigned long * rk`
- `unsigned long buf[68]`

### 12.10.1 Detailed Description

AES context structure.

Definition at line 13 of file `aes.h`.

### 12.10.2 Field Documentation

#### 12.10.2.1 `unsigned long aes_context::buf[68]`

unaligned data

Definition at line 17 of file `aes.h`.

Referenced by `aes_setkey_dec()`, and `aes_setkey_enc()`.

#### 12.10.2.2 `int aes_context::nr`

number of rounds

Definition at line 15 of file `aes.h`.

Referenced by `aes_crypt_ecb()`, `aes_setkey_dec()`, and `aes_setkey_enc()`.

#### 12.10.2.3 `unsigned long* aes_context::rk`

AES round keys

Definition at line 16 of file `aes.h`.

Referenced by `aes_crypt_ecb()`, `aes_setkey_dec()`, and `aes_setkey_enc()`.

The documentation for this struct was generated from the following file:

- `/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/aes.h`

## 12.11 agency\_s Struct Reference

The agency handle.

```
#include <libmc.h>
```

### Data Fields

- [int client](#)
- [int server](#)
- [char \\* hostName](#)
- [char \\* filename](#)
- [int portno](#)
- [int portnoc](#)
- [int initInterps](#)
- [struct mc\\_platform\\_s \\* mc\\_platform](#)
- [int default\\_agentstatus](#)
- [int threads](#)
- [int enable\\_security](#)
- [int stack\\_size](#) [MC\_THREAD\_ALL]
- [char \\* priv\\_key\\_filename](#)
- [char \\* known\\_host\\_filename](#)
- [error\\_code\\_t last\\_error](#)

### 12.11.1 Detailed Description

The agency handle.

Definition at line 224 of file libmc.h.

### 12.11.2 Field Documentation

#### 12.11.2.1 int agency\_s::client

Definition at line 225 of file libmc.h.

Referenced by MC\_Initialize().

#### 12.11.2.2 int agency\_s::default\_agentstatus

Agency default agent status

Definition at line 233 of file libmc.h.

Referenced by MC\_Initialize(), and mc\_platform\_Initialize().

#### 12.11.2.3 int agency\_s::enable\_security

Security flag

Definition at line 235 of file libmc.h.

**12.11.2.4 char\* agency\_s::filename**

Definition at line 228 of file libmc.h.

**12.11.2.5 char\* agency\_s::hostName**

Local Hostname

Definition at line 227 of file libmc.h.

Referenced by MC\_End(), and MC\_Initialize().

**12.11.2.6 int agency\_s::initInterps**

Definition at line 231 of file libmc.h.

Referenced by MC\_Initialize(), and mc\_platform\_Initialize().

**12.11.2.7 char\* agency\_s::known\_host\_filename**

Definition at line 239 of file libmc.h.

Referenced by listen\_Thread(), MC\_Initialize(), and message\_send\_Thread().

**12.11.2.8 error\_code\_t agency\_s::last\_error**

Definition at line 240 of file libmc.h.

Referenced by mc\_platform\_Initialize().

**12.11.2.9 struct mc\_platform\_s\* agency\_s::mc\_platform [read]**

Local MobileC Platform

Definition at line 232 of file libmc.h.

Referenced by MC\_AclSend(), MC\_AclSend\_chdl(), MC\_AddAgent(), MC\_AddAgent\_chdl(), MC\_-AddStationaryAgent(), MC\_Barrier(), MC\_Barrier\_chdl(), MC\_BarrierDelete(), MC\_BarrierDelete\_chdl(), MC\_BarrierInit(), MC\_BarrierInit\_chdl(), MC\_CondBroadcast(), MC\_CondBroadcast\_chdl(), MC\_CondReset(), MC\_CondReset\_chdl(), MC\_CondSignal(), MC\_CondSignal\_chdl(), MC\_CondWait(), MC\_CondWait\_chdl(), MC\_DeregisterService(), MC\_DeregisterService\_chdl(), MC\_End(), MC\_End\_chdl(), MC\_FindAgentByID(), MC\_FindAgentByID\_chdl(), MC\_FindAgentByName(), MC\_FindAgentByName\_chdl(), MC\_GetAllAgents(), MC\_HaltAgency(), MC\_HaltAgency\_chdl(), MC\_Initialize(), MC\_LoadAgentFromFile(), MC\_MainLoop(), MC\_MutexLock(), MC\_MutexLock\_chdl(), MC\_MutexUnlock(), MC\_MutexUnlock\_chdl(), MC\_RegisterService(), MC\_RegisterService\_chdl(), MC\_ResetSignal(), MC\_ResumeAgency(), MC\_ResumeAgency\_chdl(), MC\_RetrieveAgent(), MC\_RetrieveAgent\_chdl(), MC\_SearchForService(), MC\_SearchForService\_chdl(), MC\_SemaphorePost(), MC\_SemaphorePost\_chdl(), MC\_SemaphoreWait(), MC\_SemaphoreWait\_chdl(), MC\_SendAgentMigrationMessage(), MC\_SendAgentMigrationMessage\_chdl(), MC\_SendAgentMigrationMessageFile(), MC\_SendSteerCommand(), MC\_SendSteerCommand\_chdl(), MC\_SetDefaultAgentStatus(), MC\_SetDefaultAgentStatus\_chdl(), MC\_Steer(), MC\_SyncDelete(), MC\_SyncDelete\_chdl(), MC\_SyncInit(), MC\_SyncInit\_chdl(), MC\_WaitAgent(), MC\_WaitRetrieveAgent(), and MC\_WaitSignal().

**12.11.2.10 int agency\_s::portno**

Local port number

Definition at line 229 of file libmc.h.

Referenced by MC\_Initialize(), and mc\_platform\_Initialize().

**12.11.2.11 int agency\_s::portnoc**

Definition at line 230 of file libmc.h.

**12.11.2.12 char\* agency\_s::priv\_key\_filename**

Definition at line 238 of file libmc.h.

Referenced by MC\_Initialize().

**12.11.2.13 int agency\_s::server**

Definition at line 226 of file libmc.h.

Referenced by MC\_Initialize().

**12.11.2.14 int agency\_s::stack\_size[MC\_THREAD\_ALL]**

Definition at line 236 of file libmc.h.

Referenced by MC\_Initialize(), and mc\_platform\_Initialize().

**12.11.2.15 int agency\_s::threads**

flag which determines which threads to start

Definition at line 234 of file libmc.h.

Referenced by MC\_End(), MC\_Initialize(), and mc\_platform\_Initialize().

The documentation for this struct was generated from the following file:

- /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/[libmc.h](#)

## 12.12 agent\_datastate\_s Struct Reference

```
#include <agent_datastate.h>
```

### Data Fields

- char \*\* agent\_code\_ids
- char \*\* agent\_codes
- char \* agent\_code
- agent\_task\_p \* tasks
- mxml\_node\_t \* xml\_agent\_root
- mxml\_node\_t \* xml\_root
- int task\_progress
- int return\_data
- int number\_of\_tasks
- int persistent
- int init\_agent\_status
- int progress\_modifier

### 12.12.1 Detailed Description

Definition at line 41 of file agent\_datastate.h.

### 12.12.2 Field Documentation

#### 12.12.2.1 char\* agent\_datastate\_s::agent\_code

Definition at line 45 of file agent\_datastate.h.

Referenced by agent\_datastate\_Copy(), agent\_datastate\_New(), agent\_xml\_parse\_\_agent\_code(), agent\_xml\_parse\_\_tasks(), MC\_ComposeAgentS(), MC\_PrintAgentCode(), and MC\_RetrieveAgentCode().

#### 12.12.2.2 char\*\* agent\_datastate\_s::agent\_code\_ids

Definition at line 43 of file agent\_datastate.h.

Referenced by agent\_datastate\_Copy(), agent\_datastate\_Destroy(), agent\_xml\_compose\_\_agent\_code(), agent\_xml\_parse\_\_agent\_code(), agent\_xml\_parse\_\_tasks(), and MC\_ComposeAgentS().

#### 12.12.2.3 char\*\* agent\_datastate\_s::agent\_codes

Definition at line 44 of file agent\_datastate.h.

Referenced by agent\_datastate\_Copy(), agent\_datastate\_Destroy(), agent\_xml\_compose\_\_agent\_code(), agent\_xml\_parse\_\_agent\_code(), agent\_xml\_parse\_\_tasks(), and MC\_ComposeAgentS().

#### 12.12.2.4 int agent\_datastate\_s::init\_agent\_status

Definition at line 63 of file agent\_datastate.h.

Referenced by agent\_datastate\_Copy(), and agent\_datastate\_New().



### 12.12.2.5 int agent\_datastate\_s::number\_of\_tasks

Definition at line 59 of file agent\_datastate.h.

Referenced by agent\_datastate\_Copy(), agent\_datastate\_Destroy(), agent\_datastate\_New(), agent\_xml\_compose\_\_tasks(), agent\_xml\_parse\_\_agent\_code(), agent\_xml\_parse\_\_tasks(), MC\_ComposeAgentS(), MC\_GetAgentNumTasks(), MC\_GetAgentReturnData(), MC\_PrintAgentCode(), and message\_InitializeFromAgent().

### 12.12.2.6 int agent\_datastate\_s::persistent

Definition at line 62 of file agent\_datastate.h.

Referenced by acc\_MessageHandlerThread(), agent\_datastate\_Copy(), agent\_datastate\_New(), agent\_xml\_compose\_\_task(), and MC\_ComposeAgentS().

### 12.12.2.7 int agent\_datastate\_s::progress\_modifier

Definition at line 71 of file agent\_datastate.h.

Referenced by agent\_datastate\_New(), and MC\_MigrateAgent().

### 12.12.2.8 int agent\_datastate\_s::return\_data

Definition at line 56 of file agent\_datastate.h.

Referenced by agent\_datastate\_Copy(), and agent\_datastate\_New().

### 12.12.2.9 int agent\_datastate\_s::task\_progress

Definition at line 55 of file agent\_datastate.h.

Referenced by agent\_AddPersistentVariable(), agent\_datastate\_Copy(), agent\_datastate\_New(), agent\_RunChScriptThread(), agent\_xml\_compose\_\_tasks(), agent\_xml\_parse\_\_agent\_code(), agent\_xml\_parse\_\_tasks(), interpreter\_variable\_data\_InitializeFromAgent(), MC\_AgentVariableRetrieve(), MC\_AgentVariableRetrieveInfo(), MC\_AgentVariableSave(), MC\_MigrateAgent(), MC\_PrintAgentCode(), MC\_RetrieveAgentCode(), MC\_SaveData\_chdl(), and message\_InitializeFromAgent().

### 12.12.2.10 agent\_task\_p\* agent\_datastate\_s::tasks

Definition at line 48 of file agent\_datastate.h.

Referenced by agent\_AddPersistentVariable(), agent\_datastate\_Copy(), agent\_datastate\_Destroy(), agent\_datastate\_New(), agent\_xml\_compose\_\_task(), agent\_xml\_parse\_\_agent\_code(), agent\_xml\_parse\_\_data(), agent\_xml\_parse\_\_task(), agent\_xml\_parse\_\_tasks(), interpreter\_variable\_data\_InitializeFromAgent(), MC\_AgentVariableRetrieve(), MC\_AgentVariableRetrieveInfo(), MC\_AgentVariableSave(), MC\_ComposeAgentS(), MC\_GetAgentReturnData(), MC\_MigrateAgent(), MC\_SaveData\_chdl(), and message\_InitializeFromAgent().

### 12.12.2.11 mxmxml\_node\_t\* agent\_datastate\_s::xml\_agent\_root

Definition at line 51 of file agent\_datastate.h.

Referenced by `agent_datastate_New()`, `agent_Initialize()`, `agent_xml_parse()`, and `MC_GetAgentXMLString()`.

#### **12.12.2.12 `mxml_node_t* agent_datastate_s::xml_root`**

Definition at line 52 of file `agent_datastate.h`.

Referenced by `agent_datastate_Destroy()`, `agent_datastate_New()`, `agent_Initialize()`, and `agent_return_xml_parse()`.

The documentation for this struct was generated from the following file:

- `/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/agent\_datastate.h`

## 12.13 agent\_mailbox\_s Struct Reference

```
#include <agent_mailbox.h>
```

### Data Fields

- struct mail\_queue\_s \* [mail\\_queue](#)

### 12.13.1 Detailed Description

Definition at line 8 of file agent\_mailbox.h.

### 12.13.2 Field Documentation

#### 12.13.2.1 struct mail\_queue\_s\* agent\_mailbox\_s::mail\_queue [read]

Definition at line 10 of file agent\_mailbox.h.

Referenced by agent\_mailbox\_Copy(), agent\_mailbox\_Destroy(), agent\_mailbox\_New(), agent\_mailbox\_Post(), agent\_mailbox\_Retrieve(), and agent\_mailbox\_WaitRetrieve().

The documentation for this struct was generated from the following file:

- /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/[agent\\_mailbox.h](#)

## 12.14 agent\_s Struct Reference

```
#include <agent.h>
```

### Data Fields

- u\_long [id](#)
- char \* [name](#)
- u\_long [connect\\_id](#)
- time\_t [arrival\\_time](#)
- char \* [owner](#)
- char \* [home](#)
- char \* [sender](#)
- int [home\\_port](#)
- char \* [wg\\_code](#)
- char \* [agent\\_address](#)
- int [orphan](#)
- [agent\\_datastate\\_p](#) [datastate](#)
- enum [MC\\_AgentType\\_e](#) [agent\\_type](#)
- enum [MC\\_AgentStatus\\_e](#) [agent\\_status](#)
- int [return\\_data](#)
- ChInterp\_t \* [agent\\_interp](#)
- MUTEX\_T \* [run\\_lock](#)
- int [agent\\_thread\\_id](#)
- THREAD\_T [agent\\_thread](#)
- [agent\\_mailbox\\_p](#) [mailbox](#)
- int [agent\\_pipe\\_active](#)
- int [agent\\_ready\\_to\\_send](#)
- int [agent\\_pipe\\_ready\\_to\\_read](#)
- int [agent\\_script\\_ready](#)
- int [agent\\_persistent](#)
- struct [mc\\_platform\\_s](#) \* [mc\\_platform](#)
- MUTEX\_T \* [lock](#)
- int [binary](#)

### 12.14.1 Detailed Description

Definition at line 48 of file [agent.h](#).

### 12.14.2 Field Documentation

#### 12.14.2.1 char\* agent\_s::agent\_address

Definition at line 66 of file [agent.h](#).

Referenced by [agent\\_Initialize\(\)](#), and [agent\\_NewBinary\(\)](#).

### 12.14.2.2 ChInterp\_t\* agent\_s::agent\_interp

Definition at line 79 of file agent.h.

Referenced by agent\_AddPersistentVariable(), agent\_Copy(), agent\_Destroy(), agent\_RunChScriptThread(), interpreter\_variable\_data\_Initialize(), interpreter\_variable\_data\_InitializeFromAgent(), MC\_CallAgentFunc(), MC\_CallAgentFuncArg(), MC\_CallAgentFuncV(), MC\_CallAgentFuncVar(), MC\_GetAgentExecEngine(), MC\_TerminateAgent(), and MC\_TerminateAgentWG().

### 12.14.2.3 int agent\_s::agent\_persistent

Definition at line 99 of file agent.h.

Referenced by agent\_Copy().

### 12.14.2.4 int agent\_s::agent\_pipe\_active

Definition at line 95 of file agent.h.

Referenced by agent\_Initialize(), and agent\_NewBinary().

### 12.14.2.5 int agent\_s::agent\_pipe\_ready\_to\_read

Definition at line 97 of file agent.h.

Referenced by agent\_Initialize(), and agent\_NewBinary().

### 12.14.2.6 int agent\_s::agent\_ready\_to\_send

Definition at line 96 of file agent.h.

Referenced by agent\_Initialize(), and agent\_NewBinary().

### 12.14.2.7 int agent\_s::agent\_script\_ready

Definition at line 98 of file agent.h.

Referenced by agent\_Initialize(), and agent\_NewBinary().

### 12.14.2.8 enum MC\_AgentStatus\_e agent\_s::agent\_status

Definition at line 74 of file agent.h.

Referenced by acc\_MessageHandlerThread(), agent\_Copy(), agent\_Destroy(), agent\_Initialize(), agent\_NewBinary(), agent\_queue\_Flush(), agent\_RunChScript(), ams\_ManageAgentList(), ams\_Print(), AP\_QUEUE\_STD\_DEFN\_TEMPLATE(), MC\_ComposeAgentS(), MC\_GetAgentStatus(), MC\_RetrieveAgent(), MC\_SendAgentMigrationMessageFile(), and MC\_SetAgentStatus().

### 12.14.2.9 THREAD\_T agent\_s::agent\_thread

Definition at line 89 of file agent.h.

Referenced by agent\_RunChScript().

#### **12.14.2.10 int agent\_s::agent\_thread\_id**

Definition at line 85 of file agent.h.

Referenced by agent\_Initialize(), and agent\_NewBinary().

#### **12.14.2.11 enum MC\_AgentType\_e agent\_s::agent\_type**

Definition at line 73 of file agent.h.

Referenced by agent\_Copy(), agent\_Initialize(), agent\_xml\_compose\_\_message(), MC\_ComposeAgentS(), and MC\_GetAgentType().

#### **12.14.2.12 time\_t agent\_s::arrival\_time**

Definition at line 55 of file agent.h.

Referenced by agent\_Copy(), agent\_Initialize(), agent\_NewBinary(), and MC\_GetAgentArrivalTime().

#### **12.14.2.13 int agent\_s::binary**

Definition at line 105 of file agent.h.

Referenced by agent\_NewBinary(), and ams\_ManageAgentList().

#### **12.14.2.14 u\_long agent\_s::connect\_id**

Definition at line 53 of file agent.h.

Referenced by ams\_Print().

#### **12.14.2.15 agent\_datastate\_p agent\_s::datastate**

Definition at line 72 of file agent.h.

Referenced by acc\_MessageHandlerThread(), agent\_AddPersistentVariable(), agent\_Copy(), agent\_Destroy(), agent\_Initialize(), agent\_return\_xml\_parse(), agent\_RunChScriptThread(), agent\_xml\_compose\_\_agent\_code(), agent\_xml\_compose\_\_task(), agent\_xml\_compose\_\_tasks(), agent\_xml\_parse(), agent\_xml\_parse\_\_agent\_code(), agent\_xml\_parse\_\_data(), agent\_xml\_parse\_\_task(), agent\_xml\_parse\_\_tasks(), interpreter\_variable\_data\_InitializeFromAgent(), MC\_AgentVariableRetrieve(), MC\_AgentVariableRetrieveInfo(), MC\_AgentVariableSave(), MC\_ComposeAgentS(), MC\_GetAgentNumTasks(), MC\_GetAgentReturnData(), MC\_GetAgentXMLString(), MC\_MigrateAgent(), MC\_PrintAgentCode(), MC\_RetrieveAgentCode(), MC\_SaveData\_chdl(), and message\_InitializeFromAgent().

#### **12.14.2.16 char\* agent\_s::home**

Definition at line 61 of file agent.h.

Referenced by agent\_Copy(), agent\_Destroy(), agent\_Initialize(), agent\_NewBinary(), agent\_xml\_compose\_\_home(), agent\_xml\_parse\_\_home(), MC\_ComposeAgentS(), and message\_InitializeFromAgent().

#### 12.14.2.17 int agent\_s::home\_port

Definition at line 63 of file agent.h.

Referenced by agent\_Copy().

#### 12.14.2.18 u\_long agent\_s::id

Definition at line 51 of file agent.h.

Referenced by agent\_Copy(), agent\_Initialize(), agent\_NewBinary(), agent\_queue\_Flush(), agent\_RunChScriptThread(), ams\_Print(), AP\_QUEUE\_STD\_DEFN\_TEMPLATE(), MC\_GetAgentID(), and MC\_RegisterService().

#### 12.14.2.19 MUTEX\_T\* agent\_s::lock

Definition at line 103 of file agent.h.

Referenced by acc\_MessageHandlerThread(), agent\_Copy(), agent\_Destroy(), agent\_Initialize(), agent\_New(), agent\_NewBinary(), ams\_ManageAgentList(), MC\_GetAgentName(), MC\_GetAgentStatus(), MC\_PrintAgentCode(), MC\_RetrieveAgentCode(), and MC\_SetAgentStatus().

#### 12.14.2.20 agent\_mailbox\_p agent\_s::mailbox

Definition at line 92 of file agent.h.

Referenced by acc\_connection\_Thread(), agent\_Copy(), agent\_Destroy(), agent\_Initialize(), agent\_NewBinary(), MC\_AclPost(), MC\_AclRetrieve(), and MC\_AclWaitRetrieve().

#### 12.14.2.21 struct mc\_platform\_s\* agent\_s::mc\_platform [read]

Definition at line 101 of file agent.h.

Referenced by acc\_connection\_Thread(), agent\_Destroy(), agent\_Initialize(), agent\_NewBinary(), agent\_RunChScript(), agent\_RunChScriptThread(), MC\_AddAgent(), and MC\_SetAgentStatus().

#### 12.14.2.22 char\* agent\_s::name

Definition at line 52 of file agent.h.

Referenced by acc\_MessageHandlerThread(), agent\_AddPersistentVariable(), agent\_Copy(), agent\_Destroy(), agent\_queue\_Flush(), agent\_RunChScriptThread(), agent\_xml\_compose\_\_name(), agent\_xml\_parse\_\_name(), ams\_ManageAgentList(), AP\_QUEUE\_STD\_DEFN\_TEMPLATE(), MC\_AddStationaryAgent(), MC\_ComposeAgentS(), MC\_GetAgentName(), MC\_RegisterService(), and message\_InitializeFromAgent().

#### 12.14.2.23 `int agent_s::orphan`

Definition at line 68 of file agent.h.

Referenced by `agent_Copy()`, `agent_Initialize()`, `agent_NewBinary()`, `ams_ManageAgentList()`, `MC_ComposeAgentS()`, and `MC_SetAgentStatus()`.

#### 12.14.2.24 `char* agent_s::owner`

Definition at line 60 of file agent.h.

Referenced by `agent_Copy()`, `agent_Destroy()`, `agent_xml_compose__owner()`, `agent_xml_parse__owner()`, and `MC_ComposeAgentS()`.

#### 12.14.2.25 `int agent_s::return_data`

Definition at line 76 of file agent.h.

Referenced by `agent_Copy()`.

#### 12.14.2.26 `MUTEX_T* agent_s::run_lock`

Definition at line 82 of file agent.h.

Referenced by `agent_Copy()`, `agent_Destroy()`, `agent_Initialize()`, `agent_New()`, `agent_NewBinary()`, `ams_ManageAgentList()`, `interpreter_variable_data_Initialize()`, `MC_CallAgentFunc()`, `MC_CallAgentFuncArg()`, `MC_CallAgentFuncV()`, and `MC_CallAgentFuncVar()`.

#### 12.14.2.27 `char* agent_s::sender`

Definition at line 62 of file agent.h.

Referenced by `agent_Destroy()`, `agent_Initialize()`, `agent_NewBinary()`, and `agent_xml_parse__sender()`.

#### 12.14.2.28 `char* agent_s::wg_code`

Definition at line 64 of file agent.h.

Referenced by `agent_Destroy()`, `agent_xml_compose__wg_code()`, `agent_xml_parse__wg_code()`, `MC_ComposeAgentS()`, `MC_DeleteAgent()`, `MC_DeleteAgentWG()`, and `MC_TerminateAgentWG()`.

The documentation for this struct was generated from the following file:

- `/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/agent.h`



## 12.15 agent\_task\_s Struct Reference

```
#include <agent_task.h>
```

### Data Fields

- [int number\\_of\\_elements](#)
- [int size\\_of\\_element\\_array](#)
- [int persistent](#)
- [int init\\_agent\\_status](#)
- [char \\* var\\_name](#)
- [char \\* server\\_name](#)
- [char \\* code\\_id](#)
- [interpreter\\_variable\\_data\\_t \\* agent\\_return\\_data](#)
- [struct agent\\_variable\\_list\\_s \\* agent\\_variable\\_list](#)
- [char \\*\\* saved\\_variables](#)
- [int num\\_saved\\_variables](#)

### 12.15.1 Detailed Description

Definition at line 40 of file agent\_task.h.

### 12.15.2 Field Documentation

#### 12.15.2.1 [interpreter\\_variable\\_data\\_t\\* agent\\_task\\_s::agent\\_return\\_data](#)

Definition at line 52 of file agent\_task.h.

Referenced by [agent\\_task\\_Copy\(\)](#), [agent\\_task\\_Destroy\(\)](#), [agent\\_xml\\_compose\\_\\_task\(\)](#), [agent\\_xml\\_parse\\_\\_data\(\)](#), and [MC\\_GetAgentReturnData\(\)](#).

#### 12.15.2.2 [struct agent\\_variable\\_list\\_s\\* agent\\_task\\_s::agent\\_variable\\_list](#) **[read]**

Definition at line 53 of file agent\_task.h.

Referenced by [agent\\_AddPersistentVariable\(\)](#), [agent\\_task\\_Copy\(\)](#), [agent\\_task\\_Destroy\(\)](#), [agent\\_task\\_New\(\)](#), [agent\\_xml\\_compose\\_\\_task\(\)](#), [agent\\_xml\\_parse\\_\\_data\(\)](#), [MC\\_AgentVariableRetrieve\(\)](#), [MC\\_AgentVariableRetrieveInfo\(\)](#), and [MC\\_SaveData\\_chdl\(\)](#).

#### 12.15.2.3 [char\\* agent\\_task\\_s::code\\_id](#)

Definition at line 49 of file agent\_task.h.

Referenced by [agent\\_task\\_Copy\(\)](#), [agent\\_task\\_Destroy\(\)](#), [agent\\_xml\\_compose\\_\\_task\(\)](#), [agent\\_xml\\_parse\\_\\_agent\\_code\(\)](#), and [agent\\_xml\\_parse\\_\\_task\(\)](#).

#### 12.15.2.4 [int agent\\_task\\_s::init\\_agent\\_status](#)

Definition at line 45 of file agent\_task.h.

Referenced by [agent\\_task\\_Copy\(\)](#).

**12.15.2.5 int agent\_task\_s::num\_saved\_variables**

Definition at line 55 of file agent\_task.h.

Referenced by agent\_task\_Copy(), agent\_task\_New(), and MC\_AgentVariableSave().

**12.15.2.6 int agent\_task\_s::number\_of\_elements**

Definition at line 42 of file agent\_task.h.

Referenced by agent\_task\_Copy().

**12.15.2.7 int agent\_task\_s::persistent**

Definition at line 44 of file agent\_task.h.

Referenced by agent\_task\_Copy(), agent\_xml\_compose\_\_task(), agent\_xml\_parse\_\_data(), and agent\_xml\_parse\_\_task().

**12.15.2.8 char\*\* agent\_task\_s::saved\_variables**

Definition at line 54 of file agent\_task.h.

Referenced by agent\_task\_Copy(), agent\_task\_Destroy(), agent\_task\_New(), and MC\_AgentVariableSave().

**12.15.2.9 char\* agent\_task\_s::server\_name**

Definition at line 48 of file agent\_task.h.

Referenced by agent\_task\_Copy(), agent\_task\_Destroy(), agent\_xml\_compose\_\_task(), agent\_xml\_parse\_\_task(), MC\_ComposeAgentS(), MC\_MigrateAgent(), and message\_InitializeFromAgent().

**12.15.2.10 int agent\_task\_s::size\_of\_element\_array**

Definition at line 43 of file agent\_task.h.

Referenced by agent\_task\_Copy().

**12.15.2.11 char\* agent\_task\_s::var\_name**

Definition at line 47 of file agent\_task.h.

Referenced by agent\_task\_Copy(), agent\_task\_Destroy(), agent\_xml\_compose\_\_task(), agent\_xml\_parse\_\_task(), interpreter\_variable\_data\_InitializeFromAgent(), and MC\_ComposeAgentS().

The documentation for this struct was generated from the following file:

- /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/[agent\\_task.h](#)

## 12.16 agent\_thread\_arg\_s Struct Reference

```
#include <libmc.h>
```

### Data Fields

- void \* [args](#)
- struct [agent\\_s](#) \* [agent](#)
- [MCAgency\\_t](#) [attr](#)
- [THREAD\\_T](#) [thread](#)

### 12.16.1 Detailed Description

Definition at line 264 of file libmc.h.

### 12.16.2 Field Documentation

#### 12.16.2.1 struct agent\_s\* agent\_thread\_arg\_s::agent [read]

Definition at line 266 of file libmc.h.

Referenced by `MC_AddStationaryAgent()`, `MC_ComposeAgentFromFileS()`, and `MC_ComposeAgentS()`.

#### 12.16.2.2 void\* agent\_thread\_arg\_s::args

Definition at line 265 of file libmc.h.

Referenced by `MC_AddStationaryAgent()`.

#### 12.16.2.3 MCAgency\_t agent\_thread\_arg\_s::attr

Definition at line 267 of file libmc.h.

Referenced by `MC_AddStationaryAgent()`.

#### 12.16.2.4 THREAD\_T agent\_thread\_arg\_s::thread

Definition at line 268 of file libmc.h.

Referenced by `MC_AddStationaryAgent()`.

The documentation for this struct was generated from the following file:

- `/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/libmc.h`

## 12.17 AP\_GENERIC\_s Struct Reference

```
#include <ap_queue_template.h>
```

### Data Fields

- void \* [none](#)

### 12.17.1 Detailed Description

Definition at line 41 of file `ap_queue_template.h`.

### 12.17.2 Field Documentation

#### 12.17.2.1 void\* AP\_GENERIC\_s::none

Definition at line 41 of file `ap_queue_template.h`.

The documentation for this struct was generated from the following file:

- `/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/ap\_queue\_template.h`

## 12.18 arc4\_context Struct Reference

ARC4 context structure.

```
#include <arc4.h>
```

### Data Fields

- [int](#) `x`
- [int](#) `y`
- unsigned char `m` [256]

### 12.18.1 Detailed Description

ARC4 context structure.

Definition at line 10 of file arc4.h.

### 12.18.2 Field Documentation

#### 12.18.2.1 unsigned char arc4\_context::m[256]

permutation table

Definition at line 14 of file arc4.h.

Referenced by arc4\_crypt(), and arc4\_setup().

#### 12.18.2.2 int arc4\_context::x

permutation index

Definition at line 12 of file arc4.h.

Referenced by arc4\_crypt(), and arc4\_setup().

#### 12.18.2.3 int arc4\_context::y

permutation index

Definition at line 13 of file arc4.h.

Referenced by arc4\_crypt(), and arc4\_setup().

The documentation for this struct was generated from the following file:

- [/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/arc4.h](#)

## 12.19 barrier\_node\_s Struct Reference

```
#include <barrier.h>
```

### Data Fields

- `MUTEX_T * lock`
- `COND_T * cond`
- `int id`
- `int num_registered`
- `int num_waiting`

### 12.19.1 Detailed Description

Definition at line 42 of file barrier.h.

### 12.19.2 Field Documentation

#### 12.19.2.1 `COND_T* barrier_node_s::cond`

Definition at line 44 of file barrier.h.

Referenced by `barrier_node_Destroy()`, `barrier_node_Initialize()`, and `MC_Barrier()`.

#### 12.19.2.2 `int barrier_node_s::id`

Definition at line 45 of file barrier.h.

Referenced by `barrier_node_Initialize()`, `barrier_queue_Add()`, and `barrier_queue_Delete()`.

#### 12.19.2.3 `MUTEX_T* barrier_node_s::lock`

Definition at line 43 of file barrier.h.

Referenced by `barrier_node_Destroy()`, `barrier_node_Initialize()`, and `MC_Barrier()`.

#### 12.19.2.4 `int barrier_node_s::num_registered`

Definition at line 46 of file barrier.h.

Referenced by `barrier_node_Initialize()`, and `MC_Barrier()`.

#### 12.19.2.5 `int barrier_node_s::num_waiting`

Definition at line 47 of file barrier.h.

Referenced by `barrier_node_Initialize()`, and `MC_Barrier()`.

The documentation for this struct was generated from the following file:

- `/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/barrier.h`

## 12.20 barrier\_queue\_s Struct Reference

```
#include <barrier.h>
```

### Data Fields

- `RWLOCK_T * lock`
- `list_p list`
- `int size`

### 12.20.1 Detailed Description

Definition at line 51 of file barrier.h.

### 12.20.2 Field Documentation

#### 12.20.2.1 list\_p barrier\_queue\_s::list

Definition at line 54 of file barrier.h.

Referenced by `barrier_queue_Add()`, `barrier_queue_Delete()`, `barrier_queue_Destroy()`, `barrier_queue_Get()`, `barrier_queue_New()`, and `barrier_queue_Pop()`.

#### 12.20.2.2 RWLOCK\_T\* barrier\_queue\_s::lock

Definition at line 52 of file barrier.h.

Referenced by `barrier_queue_Add()`, `barrier_queue_Delete()`, `barrier_queue_Destroy()`, `barrier_queue_Get()`, and `barrier_queue_New()`.

#### 12.20.2.3 int barrier\_queue\_s::size

Definition at line 55 of file barrier.h.

Referenced by `barrier_queue_Add()`, and `barrier_queue_Delete()`.

The documentation for this struct was generated from the following file:

- `/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/barrier.h`

## 12.21 EmbeddedCh::ChBlock\_t Struct Reference

### Data Fields

- [int event\\_](#)
- [int count](#)
- [int level](#)
- [int linecurrent](#)
- [int linefuncbegin](#)
- [int linefuncend](#)
- [String source](#)
- [String funcname](#)
- [String classname](#)
- [int isconstructor](#)
- [int isdestructor](#)

### 12.21.1 Detailed Description

Definition at line 46 of file EmbedCh.cs.

### 12.21.2 Field Documentation

#### 12.21.2.1 String EmbeddedCh::ChBlock\_t::classname

Definition at line 58 of file EmbedCh.cs.

#### 12.21.2.2 int EmbeddedCh::ChBlock\_t::count

Definition at line 48 of file EmbedCh.cs.

#### 12.21.2.3 int EmbeddedCh::ChBlock\_t::event\_

Definition at line 47 of file EmbedCh.cs.

#### 12.21.2.4 String EmbeddedCh::ChBlock\_t::funcname

Definition at line 57 of file EmbedCh.cs.

#### 12.21.2.5 int EmbeddedCh::ChBlock\_t::isconstructor

Definition at line 59 of file EmbedCh.cs.

#### 12.21.2.6 int EmbeddedCh::ChBlock\_t::isdestructor

Definition at line 60 of file EmbedCh.cs.



**12.21.2.7 int EmbeddedCh::ChBlock\_t::level**

Definition at line 49 of file EmbedCh.cs.

**12.21.2.8 int EmbeddedCh::ChBlock\_t::linecurrent**

Definition at line 52 of file EmbedCh.cs.

**12.21.2.9 int EmbeddedCh::ChBlock\_t::linefuncbegin**

Definition at line 53 of file EmbedCh.cs.

**12.21.2.10 int EmbeddedCh::ChBlock\_t::linefuncend**

Definition at line 54 of file EmbedCh.cs.

**12.21.2.11 String EmbeddedCh::ChBlock\_t::source**

Definition at line 55 of file EmbedCh.cs.

The documentation for this struct was generated from the following file:

- /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/win32/EmbeddedCh.Net/EmbeddedCh.Net/[EmbedCh.cs](#)

## 12.22 EmbeddedCh::ChInfo\_t Struct Reference

### Data Fields

- String [edition](#)
- String [releasedate](#)
- String [version](#)
- uint [vermajor](#)
- uint [verminor](#)
- uint [vermicro](#)
- uint [verbuild](#)

### 12.22.1 Detailed Description

Definition at line 8 of file Ch.cs.

### 12.22.2 Field Documentation

#### 12.22.2.1 String EmbeddedCh::ChInfo\_t::edition

Definition at line 9 of file Ch.cs.

#### 12.22.2.2 String EmbeddedCh::ChInfo\_t::releasedate

Definition at line 10 of file Ch.cs.

#### 12.22.2.3 uint EmbeddedCh::ChInfo\_t::verbuild

Definition at line 15 of file Ch.cs.

#### 12.22.2.4 uint EmbeddedCh::ChInfo\_t::vermajor

Definition at line 12 of file Ch.cs.

#### 12.22.2.5 uint EmbeddedCh::ChInfo\_t::vermicro

Definition at line 14 of file Ch.cs.

#### 12.22.2.6 uint EmbeddedCh::ChInfo\_t::verminor

Definition at line 13 of file Ch.cs.

#### 12.22.2.7 String EmbeddedCh::ChInfo\_t::version

Definition at line 11 of file Ch.cs.

The documentation for this struct was generated from the following file:

- </home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/win32/EmbeddedCh.Net/EmbeddedCh.Net/Ch.cs>

## 12.23 EmbeddedCh::ChInterp Class Reference

### Public Member Functions

- delegate void [ChCallback](#) (IntPtr [interp](#), ref [ChBlock\\_t](#) calldata, IntPtr clientdata)
- [ChInterp](#) ()
- [int Initialize](#) ()
- [int RunScript](#) (String[] argv)
- [int AppendRunScript](#) (String argv)
- [int AppendRunScriptFile](#) (String filename)
- [int InitGlobalVar](#) (Int32 flag)
- [int ParseScript](#) (String[] argv)
- [int ExecScript](#) (String progname)
- [int ExecScriptM](#) (String progname)
- [int RunScriptM](#) (String[] argv)
- [int AppendParseScript](#) (String code)
- [int AppendParseScriptFile](#) (String filename)
- [int End](#) ()
- [int ExprCalc](#) (String expr, [ChType\\_t](#) datatype, IntPtr result)
- [int ExprEval](#) (String expr)
- [int ExprParse](#) (String expr)
- IntPtr [ExprValue](#) (String expr, IntPtr result)
- [int DeleteExprValue](#) (IntPtr vn)
- Int32 [SetVar](#) (String name, [ChType\\_t](#) atype)
- Int32 [Close](#) (Int32 fildes)
- Int32 [Reopen](#) (String filename, String mode, Int32 fildes)
- Int32 [Flush](#) (Int32 fildes)
- Int32 [Abort](#) ()
- Int32 [DeclareVar](#) (String declaration)
- Int32 [DeclareTypedef](#) (String name)
- Int32 [DeclareFunc](#) (String funcprototype, IntPtr funcptr)
- Int32 [SetGlobalUserData](#) (IntPtr userdata)
- IntPtr [GetGlobalUserData](#) ()
- Int32 [AddCallback](#) (Int32 \_event, ChCallback callback, IntPtr clientdata, Int32 count)
- Int32 [ChangeStack](#) (Int32 level, ref [ChBlock\\_t](#) calldata)
- Int32 [StackLevel](#) (ref Int32 clevel, ref Int32 hlevel)
- String [StackName](#) (Int32 level, ref Int32 isfunc, String[] classname)
- Int32 [GlobalSymbolTotalNum](#) ()
- Int32 [GlobalSymbolIndexByName](#) (String name)
- IntPtr [GlobalSymbolAddrByIndex](#) (Int32 index)
- String [GlobalSymbolNameByIndex](#) (Int32 index)
- [ChType\\_t DataType](#) (String expr)
- Int32 [DataSize](#) (String expr)
- [ChType\\_t ArrayType](#) (String expr)
- Int32 [ArrayDim](#) (String expr)
- Int32 [ArrayExtent](#) (String expr, Int32 index)
- Int32 [ArrayNum](#) (String expr)
- [ChVarType\\_t VarType](#) (String name)
- [ChFuncType\\_t FuncType](#) (String name)
- Int32 [IsFuncVarArg](#) (String name)

- Int32 [FuncArgNum](#) (String name)
- IntPtr [SymbolAddrByName](#) (String name)
- Int32 [SymbolTotalNum](#) ()
- Int32 [SymbolIndexByName](#) (String name)
- IntPtr [SymbolAddrByIndex](#) (Int32 index)
- String [SymbolNameByIndex](#) (Int32 index)
- IntPtr [UserDefinedTag](#) (String expr)
- Int32 [UserDefinedInfo](#) (IntPtr udtag, ref [ChUserDefinedInfo\\_t](#) udtinfo)
- Int32 [UserDefinedMemInfoByName](#) (IntPtr udtag, String memname, ref [ChMemInfo\\_t](#) meminfo)
- Int32 [UserDefinedMemInfoByIndex](#) (IntPtr udtag, Int32 index, ref [ChMemInfo\\_t](#) meminfo)
- String [UserDefinedName](#) (String name)
- Int32 [UserDefinedSize](#) (String name)
- [ChType\\_t](#) [FuncArgDataType](#) (String funcname, Int32 argnum)
- [ChType\\_t](#) [FuncArgArrayType](#) (String funcname, Int32 argnum)
- Int32 [FuncArgArrayDim](#) (String funcname, Int32 argnum)
- Int32 [FuncArgArrayExtent](#) (String funcname, Int32 argnum, Int32 index)
- Int32 [FuncArgArrayNum](#) (String funcname, Int32 argnum)
- Int32 [FuncArgIsFunc](#) (String funcname, Int32 argnum)
- Int32 [FuncArgIsFuncVarArg](#) (String funcname, Int32 argnum)
- Int32 [FuncArgFuncArgNum](#) (String funcname, Int32 argnum)
- String [FuncArgUserDefinedName](#) (String funcname, Int32 argnum)
- Int32 [FuncArgUserDefinedSize](#) (String funcname, Int32 argnum)

## Data Fields

- const String [chdll](#) = "embedchdll.dll"

## Properties

- bool [UseOptions](#) [get, set]
- [ChShellType](#) [ShellType](#) [get, set]
- String [ChHome](#) [get, set]

## Private Member Functions

- static Int32 [\\_Ch\\_Initialize](#) (ref IntPtr interpp, ref [ChOptions\\_t](#) option)
- static Int32 [\\_Ch\\_Initialize](#) (ref IntPtr interpp, IntPtr option)
- static Int32 [\\_Ch\\_InitGlobalVar](#) (IntPtr [interp](#), Int32 flag)
- static Int32 [\\_Ch\\_ParseScript](#) (IntPtr [interp](#), String[] argv)
- static Int32 [\\_Ch\\_ExecScript](#) (IntPtr [interp](#), String progname)
- static Int32 [\\_Ch\\_ExecScriptM](#) (IntPtr [interp](#), String progname)
- static Int32 [\\_Ch\\_RunScript](#) (IntPtr [interp](#), String[] argv)
- static Int32 [\\_Ch\\_RunScriptM](#) (IntPtr [interp](#), String[] argv)
- static Int32 [\\_Ch\\_AppendParseScript](#) (IntPtr [interp](#), String code)
- static Int32 [\\_Ch\\_AppendParseScriptFile](#) (IntPtr [interp](#), String filename)
- static Int32 [\\_Ch\\_AppendRunScript](#) (IntPtr [interp](#), String code)
- static Int32 [\\_Ch\\_AppendRunScriptFile](#) (IntPtr [interp](#), String filename)
- static Int32 [\\_Ch\\_ExprCalc](#) (IntPtr [interp](#), String expr, [ChType\\_t](#) datatype, IntPtr result)
- static Int32 [\\_Ch\\_ExprEval](#) (IntPtr [interp](#), String expr)

- static Int32 [\\_Ch\\_ExprParse](#) (IntPtr [interp](#), String expr)
- static IntPtr [\\_Ch\\_ExprValue](#) (IntPtr [interp](#), String expr, IntPtr result)
- static Int32 [\\_Ch\\_DeleteExprValue](#) (IntPtr [interp](#), IntPtr vn)
- static Int32 [\\_Ch\\_SetVar](#) (IntPtr [interp](#), String name, [ChType\\_t](#) atype)
- static Int32 [\\_Ch\\_Close](#) (IntPtr [interp](#), Int32 fildes)
- static Int32 [\\_Ch\\_Reopen](#) (IntPtr [interp](#), String filename, String mode, Int32 fildes)
- static Int32 [\\_Ch\\_Flush](#) (IntPtr [interp](#), Int32 fildes)
- static Int32 [\\_Ch\\_End](#) (IntPtr [interp](#))
- static Int32 [\\_Ch\\_Abort](#) (IntPtr [interp](#))
- static Int32 [\\_Ch\\_DeclareVar](#) (IntPtr [interp](#), String declaration)
- static Int32 [\\_Ch\\_DeclareTypedef](#) (IntPtr [interp](#), String name)
- static Int32 [\\_Ch\\_DeclareFunc](#) (IntPtr [interp](#), String funcprototype, IntPtr funcptr)
- static Int32 [\\_Ch\\_SetGlobalUserData](#) (IntPtr [interp](#), IntPtr userdata)
- static IntPtr [\\_Ch\\_GetGlobalUserData](#) (IntPtr [interp](#))
- static Int32 [\\_Ch\\_AddCallback](#) (IntPtr [interp](#), Int32 \_event, ChCallback callback, IntPtr clientdata, Int32 count)
- static Int32 [\\_Ch\\_ChangeStack](#) (IntPtr [interp](#), Int32 level, ref [ChBlock\\_t](#) calldata)
- static Int32 [\\_Ch\\_StackLevel](#) (IntPtr [interp](#), ref Int32 clevel, ref Int32 hlevel)
- static String [\\_Ch\\_StackName](#) (IntPtr [interp](#), Int32 level, ref Int32 isfunc, String[ ] classname)
- static Int32 [\\_Ch\\_GlobalSymbolTotalNum](#) (IntPtr [interp](#))
- static Int32 [\\_Ch\\_GlobalSymbolIndexByName](#) (IntPtr [interp](#), String name)
- static IntPtr [\\_Ch\\_GlobalSymbolAddrByIndex](#) (IntPtr [interp](#), Int32 index)
- static String [\\_Ch\\_GlobalSymbolNameByIndex](#) (IntPtr [interp](#), Int32 index)
- static [ChType\\_t](#) [\\_Ch\\_DataType](#) (IntPtr [interp](#), String expr)
- static Int32 [\\_Ch\\_DataSize](#) (IntPtr [interp](#), String expr)
- static [ChType\\_t](#) [\\_Ch\\_ArrayType](#) (IntPtr [interp](#), String expr)
- static Int32 [\\_Ch\\_ArrayDim](#) (IntPtr [interp](#), String expr)
- static Int32 [\\_Ch\\_ArrayExtent](#) (IntPtr [interp](#), String expr, Int32 index)
- static Int32 [\\_Ch\\_ArrayNum](#) (IntPtr [interp](#), String expr)
- static [ChVarType\\_t](#) [\\_Ch\\_VarType](#) (IntPtr [interp](#), String name)
- static [ChFuncType\\_t](#) [\\_Ch\\_FuncType](#) (IntPtr [interp](#), String name)
- static Int32 [\\_Ch\\_IsFuncVarArg](#) (IntPtr [interp](#), String name)
- static Int32 [\\_Ch\\_FuncArgNum](#) (IntPtr [interp](#), String name)
- static IntPtr [\\_Ch\\_SymbolAddrByName](#) (IntPtr [interp](#), String name)
- static Int32 [\\_Ch\\_SymbolTotalNum](#) (IntPtr [interp](#))
- static Int32 [\\_Ch\\_SymbolIndexByName](#) (IntPtr [interp](#), String name)
- static IntPtr [\\_Ch\\_SymbolAddrByIndex](#) (IntPtr [interp](#), Int32 index)
- static String [\\_Ch\\_SymbolNameByIndex](#) (IntPtr [interp](#), Int32 index)
- static IntPtr [\\_Ch\\_UserDefinedTag](#) (IntPtr [interp](#), String expr)
- static Int32 [\\_Ch\\_UserDefinedInfo](#) (IntPtr [interp](#), IntPtr udtag, ref [ChUserDefinedInfo\\_t](#) udinfo)
- static Int32 [\\_Ch\\_UserDefinedMemInfoByName](#) (IntPtr [interp](#), IntPtr udtag, String memname, ref [ChMemInfo\\_t](#) meminfo)
- static Int32 [\\_Ch\\_UserDefinedMemInfoByIndex](#) (IntPtr [interp](#), IntPtr udtag, Int32 index, ref [ChMemInfo\\_t](#) meminfo)
- static String [\\_Ch\\_UserDefinedName](#) (IntPtr [interp](#), String name)
- static Int32 [\\_Ch\\_UserDefinedSize](#) (IntPtr [interp](#), String name)
- static [ChType\\_t](#) [\\_Ch\\_FuncArgDataType](#) (IntPtr [interp](#), String funcname, Int32 argnum)
- static [ChType\\_t](#) [\\_Ch\\_FuncArgArrayType](#) (IntPtr [interp](#), String funcname, Int32 argnum)
- static Int32 [\\_Ch\\_FuncArgArrayDim](#) (IntPtr [interp](#), String funcname, Int32 argnum)
- static Int32 [\\_Ch\\_FuncArgArrayExtent](#) (IntPtr [interp](#), String funcname, Int32 argnum, Int32 index)

- static Int32 `_Ch_FuncArgArrayNum` (IntPtr *interp*, String funcname, Int32 argnum)
- static Int32 `_Ch_FuncArgIsFunc` (IntPtr *interp*, String funcname, Int32 argnum)
- static Int32 `_Ch_FuncArgIsFuncVarArg` (IntPtr *interp*, String funcname, Int32 argnum)
- static Int32 `_Ch_FuncArgFuncArgNum` (IntPtr *interp*, String funcname, Int32 argnum)
- static String `_Ch_FuncArgUserDefinedName` (IntPtr *interp*, String funcname, Int32 argnum)
- static Int32 `_Ch_FuncArgUserDefinedSize` (IntPtr *interp*, String funcname, Int32 argnum)
- static String `_Ch_Home` (IntPtr *interp*)
- static Int32 `_Ch_Version` (IntPtr *interp*, ChInfo\_t \*info)
- static IntPtr `_Ch_GlobalSymbolAddrByName` (IntPtr *interp*, String name)
- static IntPtr `_Ch_SymbolAddrByName` (IntPtr *interp*, String name)
- static Int32 `_Ch_CallFuncByAddrv` (IntPtr *interp*, IntPtr fptr, IntPtr retval, va\_list ap)
- static Int32 `_Ch_CallFuncByNamev` (IntPtr *interp*, String name, IntPtr retval, va\_list ap)
- static Int32 `_Ch_CallFuncByNameVar` (IntPtr *interp*, String name, IntPtr retval, ChVaList\_t arglist)

## Private Attributes

- IntPtr *interp* = IntPtr.Zero
- ChOptions\_t *options* = new ChOptions\_t()
- bool *useOptions* = false

### 12.23.1 Detailed Description

Definition at line 9 of file ChInterp.cs.

### 12.23.2 Constructor & Destructor Documentation

#### 12.23.2.1 EmbeddedCh::ChInterp::ChInterp () [inline]

Definition at line 25 of file ChInterp.cs.

### 12.23.3 Member Function Documentation

#### 12.23.3.1 static Int32 EmbeddedCh::ChInterp::\_Ch\_Abort (IntPtr *interp*) [private]

Referenced by Abort().

#### 12.23.3.2 static Int32 EmbeddedCh::ChInterp::\_Ch\_AddCallback (IntPtr *interp*, Int32 *\_event*, ChCallback *callback*, IntPtr *clientdata*, Int32 *count*) [private]

Referenced by AddCallback().

#### 12.23.3.3 static Int32 EmbeddedCh::ChInterp::\_Ch\_AppendParseScript (IntPtr *interp*, String *code*) [private]

Referenced by AppendParseScript().

**12.23.3.4** `static Int32 EmbeddedCh::ChInterp::_Ch_AppendParseScriptFile (IntPtr interp, String filename) [private]`

Referenced by AppendParseScriptFile().

**12.23.3.5** `static Int32 EmbeddedCh::ChInterp::_Ch_AppendRunScript (IntPtr interp, String code) [private]`

Referenced by AppendRunScript().

**12.23.3.6** `static Int32 EmbeddedCh::ChInterp::_Ch_AppendRunScriptFile (IntPtr interp, String filename) [private]`

Referenced by AppendRunScriptFile().

**12.23.3.7** `static Int32 EmbeddedCh::ChInterp::_Ch_ArrayDim (IntPtr interp, String expr) [private]`

Referenced by ArrayDim().

**12.23.3.8** `static Int32 EmbeddedCh::ChInterp::_Ch_ArrayExtent (IntPtr interp, String expr, Int32 index) [private]`

Referenced by ArrayExtent().

**12.23.3.9** `static Int32 EmbeddedCh::ChInterp::_Ch_ArrayNum (IntPtr interp, String expr) [private]`

Referenced by ArrayNum().

**12.23.3.10** `static ChType_t EmbeddedCh::ChInterp::_Ch_ArrayType (IntPtr interp, String expr) [private]`

Referenced by ArrayType().

**12.23.3.11** `static Int32 EmbeddedCh::ChInterp::_Ch_CallFuncByAddrv (IntPtr interp, IntPtr fptr, IntPtr retval, va_list ap) [private]`

**12.23.3.12** `static Int32 EmbeddedCh::ChInterp::_Ch_CallFuncByNamev (IntPtr interp, String name, IntPtr retval, va_list ap) [private]`

**12.23.3.13** `static Int32 EmbeddedCh::ChInterp::_Ch_CallFuncByNameVar (IntPtr interp, String name, IntPtr retval, ChVaList_t arglist) [private]`

**12.23.3.14** `static Int32 EmbeddedCh::ChInterp::_Ch_ChangeStack (IntPtr interp, Int32 level, ref ChBlock_t calldata) [private]`

Referenced by ChangeStack().



**12.23.3.15** `static Int32 EmbeddedCh::ChInterp::_Ch_Close (IntPtr interp, Int32 fildes) [private]`

Referenced by Close().

**12.23.3.16** `static Int32 EmbeddedCh::ChInterp::_Ch_DataSize (IntPtr interp, String expr) [private]`

Referenced by DataSize().

**12.23.3.17** `static ChType_t EmbeddedCh::ChInterp::_Ch_DataType (IntPtr interp, String expr) [private]`

Referenced by DataType().

**12.23.3.18** `static Int32 EmbeddedCh::ChInterp::_Ch_DeclareFunc (IntPtr interp, String funcprototype, IntPtr funcptr) [private]`

Referenced by DeclareFunc().

**12.23.3.19** `static Int32 EmbeddedCh::ChInterp::_Ch_DeclareTypedef (IntPtr interp, String name) [private]`

Referenced by DeclareTypedef().

**12.23.3.20** `static Int32 EmbeddedCh::ChInterp::_Ch_DeclareVar (IntPtr interp, String declaration) [private]`

Referenced by DeclareVar().

**12.23.3.21** `static Int32 EmbeddedCh::ChInterp::_Ch_DeleteExprValue (IntPtr interp, IntPtr vn) [private]`

Referenced by DeleteExprValue().

**12.23.3.22** `static Int32 EmbeddedCh::ChInterp::_Ch_End (IntPtr interp) [private]`

Referenced by End().

**12.23.3.23** `static Int32 EmbeddedCh::ChInterp::_Ch_ExecScript (IntPtr interp, String progname) [private]`

Referenced by ExecScript().

**12.23.3.24** `static Int32 EmbeddedCh::ChInterp::_Ch_ExecScriptM (IntPtr interp, String progname) [private]`

Referenced by ExecScriptM().

**12.23.3.25** `static Int32 EmbeddedCh::ChInterp::_Ch_ExprCalc (IntPtr interp, String expr, ChType_t datatype, IntPtr result) [private]`

Referenced by ExprCalc().

**12.23.3.26** `static Int32 EmbeddedCh::ChInterp::_Ch_ExprEval (IntPtr interp, String expr) [private]`

Referenced by ExprEval().

**12.23.3.27** `static Int32 EmbeddedCh::ChInterp::_Ch_ExprParse (IntPtr interp, String expr) [private]`

Referenced by ExprParse().

**12.23.3.28** `static IntPtr EmbeddedCh::ChInterp::_Ch_ExprValue (IntPtr interp, String expr, IntPtr result) [private]`

Referenced by ExprValue().

**12.23.3.29** `static Int32 EmbeddedCh::ChInterp::_Ch_Flush (IntPtr interp, Int32 filde) [private]`

Referenced by Flush().

**12.23.3.30** `static Int32 EmbeddedCh::ChInterp::_Ch_FuncArgArrayDim (IntPtr interp, String funcname, Int32 argnum) [private]`

Referenced by FuncArgArrayDim().

**12.23.3.31** `static Int32 EmbeddedCh::ChInterp::_Ch_FuncArgArrayExtent (IntPtr interp, String funcname, Int32 argnum, Int32 index) [private]`

Referenced by FuncArgArrayExtent().

**12.23.3.32** `static Int32 EmbeddedCh::ChInterp::_Ch_FuncArgArrayNum (IntPtr interp, String funcname, Int32 argnum) [private]`

Referenced by FuncArgArrayNum().

**12.23.3.33** `static ChType_t EmbeddedCh::ChInterp::_Ch_FuncArgArrayType (IntPtr interp, String funcname, Int32 argnum) [private]`

Referenced by FuncArgArrayType().

**12.23.3.34** `static ChType_t EmbeddedCh::ChInterp::_Ch_FuncArgDataType (IntPtr interp, String funcname, Int32 argnum) [private]`

Referenced by FuncArgDataType().

**12.23.3.35** `static Int32 EmbeddedCh::ChInterp::_Ch_FuncArgFuncArgNum (IntPtr interp, String funcname, Int32 argnum) [private]`

Referenced by FuncArgFuncArgNum().

**12.23.3.36** `static Int32 EmbeddedCh::ChInterp::_Ch_FuncArgIsFunc (IntPtr interp, String funcname, Int32 argnum) [private]`

Referenced by FuncArgIsFunc().

**12.23.3.37** `static Int32 EmbeddedCh::ChInterp::_Ch_FuncArgIsFuncVarArg (IntPtr interp, String funcname, Int32 argnum) [private]`

Referenced by FuncArgIsFuncVarArg().

**12.23.3.38** `static Int32 EmbeddedCh::ChInterp::_Ch_FuncArgNum (IntPtr interp, String name) [private]`

Referenced by FuncArgNum().

**12.23.3.39** `static String EmbeddedCh::ChInterp::_Ch_FuncArgUserDefinedName (IntPtr interp, String funcname, Int32 argnum) [private]`

Referenced by FuncArgUserDefinedName().

**12.23.3.40** `static Int32 EmbeddedCh::ChInterp::_Ch_FuncArgUserDefinedSize (IntPtr interp, String funcname, Int32 argnum) [private]`

Referenced by FuncArgUserDefinedSize().

**12.23.3.41** `static ChFuncType_t EmbeddedCh::ChInterp::_Ch_FuncType (IntPtr interp, String name) [private]`

Referenced by FuncType().

**12.23.3.42** `static IntPtr EmbeddedCh::ChInterp::_Ch_GetGlobalUserData (IntPtr interp) [private]`

Referenced by GetGlobalUserData().

**12.23.3.43** `static IntPtr EmbeddedCh::ChInterp::_Ch_GlobalSymbolAddrByIndex (IntPtr interp, Int32 index) [private]`

Referenced by GlobalSymbolAddrByIndex().

**12.23.3.44** `static IntPtr EmbeddedCh::ChInterp::_Ch_GlobalSymbolAddrByName (IntPtr interp, String name) [private]`

**12.23.3.45** `static Int32 EmbeddedCh::ChInterp::_Ch_GlobalSymbolIndexByName (IntPtr interp, String name) [private]`

Referenced by GlobalSymbolIndexByName().

**12.23.3.46** `static String EmbeddedCh::ChInterp::_Ch_GlobalSymbolNameByIndex (IntPtr interp, Int32 index) [private]`

Referenced by GlobalSymbolNameByIndex().

**12.23.3.47** `static Int32 EmbeddedCh::ChInterp::_Ch_GlobalSymbolTotalNum (IntPtr interp) [private]`

Referenced by GlobalSymbolTotalNum().

**12.23.3.48** `static String EmbeddedCh::ChInterp::_Ch_Home (IntPtr interp) [private]`

**12.23.3.49** `static Int32 EmbeddedCh::ChInterp::_Ch_InitGlobalVar (IntPtr interp, Int32 flag) [private]`

Referenced by InitGlobalVar().

**12.23.3.50** `static Int32 EmbeddedCh::ChInterp::_Ch_Initialize (ref IntPtr interp, IntPtr option) [private]`

**12.23.3.51** `static Int32 EmbeddedCh::ChInterp::_Ch_Initialize (ref IntPtr interp, ref ChOptions_t option) [private]`

Referenced by Initialize().

**12.23.3.52** `static Int32 EmbeddedCh::ChInterp::_Ch_IsFuncVarArg (IntPtr interp, String name) [private]`

Referenced by IsFuncVarArg().

**12.23.3.53** `static Int32 EmbeddedCh::ChInterp::_Ch_ParseScript (IntPtr interp, String[] argv) [private]`

Referenced by ParseScript().

**12.23.3.54** `static Int32 EmbeddedCh::ChInterp::_Ch_Reopen (IntPtr interp, String filename, String mode, Int32 fildev) [private]`

Referenced by Reopen().

**12.23.3.55** `static Int32 EmbeddedCh::ChInterp::_Ch_RunScript (IntPtr interp, String[ ] argv) [private]`

Referenced by RunScript().

**12.23.3.56** `static Int32 EmbeddedCh::ChInterp::_Ch_RunScriptM (IntPtr interp, String[ ] argv) [private]`

Referenced by RunScriptM().

**12.23.3.57** `static Int32 EmbeddedCh::ChInterp::_Ch_SetGlobalUserData (IntPtr interp, IntPtr userdata) [private]`

Referenced by SetGlobalUserData().

**12.23.3.58** `static Int32 EmbeddedCh::ChInterp::_Ch_SetVar (IntPtr interp, String name, ChType type) [private]`

Referenced by SetVar().

**12.23.3.59** `static Int32 EmbeddedCh::ChInterp::_Ch_StackLevel (IntPtr interp, ref Int32 clevel, ref Int32 hlevel) [private]`

Referenced by StackLevel().

**12.23.3.60** `static String EmbeddedCh::ChInterp::_Ch_StackName (IntPtr interp, Int32 level, ref Int32 isfunc, String[ ] classname) [private]`

Referenced by StackName().

**12.23.3.61** `static IntPtr EmbeddedCh::ChInterp::_Ch_SymbolAddrByIndex (IntPtr interp, Int32 index) [private]`

Referenced by SymbolAddrByIndex().

**12.23.3.62** `static IntPtr EmbeddedCh::ChInterp::_Ch_SymbolAddrByName (IntPtr interp, String name) [private]`

**12.23.3.63** `static IntPtr EmbeddedCh::ChInterp::_Ch_SymbolAddrByName (IntPtr interp, String name) [private]`

Referenced by SymbolAddrByName().

**12.23.3.64** `static Int32 EmbeddedCh::ChInterp::_Ch_SymbolIndexByName (IntPtr interp, String name) [private]`

Referenced by SymbolIndexByName().

**12.23.3.65** `static String EmbeddedCh::ChInterp::_Ch_SymbolNameByIndex (IntPtr interp, Int32 index) [private]`

Referenced by SymbolNameByIndex().

**12.23.3.66** `static Int32 EmbeddedCh::ChInterp::_Ch_SymbolTotalNum (IntPtr interp) [private]`

Referenced by SymbolTotalNum().

**12.23.3.67** `static Int32 EmbeddedCh::ChInterp::_Ch_UserDefinedInfo (IntPtr interp, IntPtr udtag, ref ChUserDefinedInfo_t udinfo) [private]`

Referenced by UserDefinedInfo().

**12.23.3.68** `static Int32 EmbeddedCh::ChInterp::_Ch_UserDefinedMemInfoByIndex (IntPtr interp, IntPtr udtag, Int32 index, ref ChMemInfo_t meminfo) [private]`

Referenced by UserDefinedMemInfoByIndex().

**12.23.3.69** `static Int32 EmbeddedCh::ChInterp::_Ch_UserDefinedMemInfoByName (IntPtr interp, IntPtr udtag, String memname, ref ChMemInfo_t meminfo) [private]`

Referenced by UserDefinedMemInfoByName().

**12.23.3.70** `static String EmbeddedCh::ChInterp::_Ch_UserDefinedName (IntPtr interp, String name) [private]`

Referenced by UserDefinedName().

**12.23.3.71** `static Int32 EmbeddedCh::ChInterp::_Ch_UserDefinedSize (IntPtr interp, String name) [private]`

Referenced by UserDefinedSize().

**12.23.3.72** `static IntPtr EmbeddedCh::ChInterp::_Ch_UserDefinedTag (IntPtr interp, String expr) [private]`

Referenced by UserDefinedTag().

**12.23.3.73** `static ChVarType_t EmbeddedCh::ChInterp::_Ch_VarType (IntPtr interp, String name) [private]`

Referenced by VarType().

**12.23.3.74** `static Int32 EmbeddedCh::ChInterp::_Ch_Version (IntPtr interp, ChInfo_t * info) [private]`

**12.23.3.75** `Int32 EmbeddedCh::ChInterp::Abort () [inline]`

Definition at line 183 of file ChInterp.cs.

References \_Ch\_Abort(), and interp.

**12.23.3.76** `Int32 EmbeddedCh::ChInterp::AddCallback (Int32 _event, ChCallback callback, IntPtr clientdata, Int32 count) [inline]`

Definition at line 213 of file ChInterp.cs.

References \_Ch\_AddCallback(), and interp.

**12.23.3.77** `int EmbeddedCh::ChInterp::AppendParseScript (String code) [inline]`

Definition at line 113 of file ChInterp.cs.

References \_Ch\_AppendParseScript(), and interp.

**12.23.3.78** `int EmbeddedCh::ChInterp::AppendParseScriptFile (String filename) [inline]`

Definition at line 118 of file ChInterp.cs.

References \_Ch\_AppendParseScriptFile(), and interp.

**12.23.3.79** `int EmbeddedCh::ChInterp::AppendRunScript (String argv) [inline]`

Definition at line 78 of file ChInterp.cs.

References \_Ch\_AppendRunScript(), and interp.

Referenced by Program1::Program::Main().

**12.23.3.80** `int EmbeddedCh::ChInterp::AppendRunScriptFile (String filename) [inline]`

Definition at line 83 of file ChInterp.cs.

References \_Ch\_AppendRunScriptFile(), and interp.

Referenced by Program1::Program::Main().

**12.23.3.81** `Int32 EmbeddedCh::ChInterp::ArrayDim (String expr) [inline]`

Definition at line 269 of file ChInterp.cs.

References \_Ch\_ArrayDim(), and interp.

**12.23.3.82 Int32 EmbeddedCh::ChInterp::ArrayExtent (String *expr*, Int32 *index*) [inline]**

Definition at line 274 of file ChInterp.cs.

References `_Ch_ArrayExtent()`, and `interp`.

**12.23.3.83 Int32 EmbeddedCh::ChInterp::ArrayNum (String *expr*) [inline]**

Definition at line 279 of file ChInterp.cs.

References `_Ch_ArrayNum()`, and `interp`.

**12.23.3.84 ChType\_t EmbeddedCh::ChInterp::ArrayType (String *expr*) [inline]**

Definition at line 264 of file ChInterp.cs.

References `_Ch_ArrayType()`, and `interp`.

**12.23.3.85 Int32 EmbeddedCh::ChInterp::ChangeStack (Int32 *level*, ref ChBlock\_t *calldata*) [inline]**

Definition at line 219 of file ChInterp.cs.

References `_Ch_ChangeStack()`, and `interp`.

**12.23.3.86 delegate void EmbeddedCh::ChInterp::ChCallback (IntPtr *interp*, ref ChBlock\_t *calldata*, IntPtr *clientdata*)****12.23.3.87 Int32 EmbeddedCh::ChInterp::Close (Int32 *fildes*) [inline]**

Definition at line 168 of file ChInterp.cs.

References `_Ch_Close()`, and `interp`.

**12.23.3.88 Int32 EmbeddedCh::ChInterp::DataSize (String *expr*) [inline]**

Definition at line 259 of file ChInterp.cs.

References `_Ch_DataSize()`, and `interp`.

**12.23.3.89 ChType\_t EmbeddedCh::ChInterp::DataType (String *expr*) [inline]**

Definition at line 254 of file ChInterp.cs.

References `_Ch_DataType()`, and `interp`.

**12.23.3.90 Int32 EmbeddedCh::ChInterp::DeclareFunc (String *funcprototype*, IntPtr *funcptr*) [inline]**

Definition at line 198 of file ChInterp.cs.

References `_Ch_DeclareFunc()`, and `interp`.



**12.23.3.91 Int32 EmbeddedCh::ChInterp::DeclareTypedef (String *name*) [inline]**

Definition at line 193 of file ChInterp.cs.

References `_Ch_DeclareTypedef()`, and `interp`.

**12.23.3.92 Int32 EmbeddedCh::ChInterp::DeclareVar (String *declaration*) [inline]**

Definition at line 188 of file ChInterp.cs.

References `_Ch_DeclareVar()`, and `interp`.

**12.23.3.93 int EmbeddedCh::ChInterp::DeleteExprValue (IntPtr *vn*) [inline]**

Definition at line 153 of file ChInterp.cs.

References `_Ch_DeleteExprValue()`, and `interp`.

**12.23.3.94 int EmbeddedCh::ChInterp::End () [inline]**

Definition at line 123 of file ChInterp.cs.

References `_Ch_End()`, and `interp`.

Referenced by `Program1::Program::Main()`.

**12.23.3.95 int EmbeddedCh::ChInterp::ExecScript (String *programe*) [inline]**

Definition at line 98 of file ChInterp.cs.

References `_Ch_ExecScript()`, and `interp`.

**12.23.3.96 int EmbeddedCh::ChInterp::ExecScriptM (String *programe*) [inline]**

Definition at line 103 of file ChInterp.cs.

References `_Ch_ExecScriptM()`, and `interp`.

**12.23.3.97 int EmbeddedCh::ChInterp::ExprCalc (String *expr*, ChType\_t *datatype*, IntPtr *result*) [inline]**

Definition at line 131 of file ChInterp.cs.

References `_Ch_ExprCalc()`, and `interp`.

**12.23.3.98 int EmbeddedCh::ChInterp::ExprEval (String *expr*) [inline]**

Definition at line 136 of file ChInterp.cs.

References `_Ch_ExprEval()`, and `interp`.

**12.23.3.99 int EmbeddedCh::ChInterp::ExprParse (String *expr*) [inline]**

Definition at line 141 of file ChInterp.cs.

References `_Ch_ExprParse()`, and `interp`.

**12.23.3.100 IntPtr EmbeddedCh::ChInterp::ExprValue (String *expr*, IntPtr *result*) [inline]**

Definition at line 147 of file ChInterp.cs.

References `_Ch_ExprValue()`, and `interp`.

**12.23.3.101 Int32 EmbeddedCh::ChInterp::Flush (Int32 *fildev*) [inline]**

Definition at line 178 of file ChInterp.cs.

References `_Ch_Flush()`, and `interp`.

**12.23.3.102 Int32 EmbeddedCh::ChInterp::FuncArgArrayDim (String *funcname*, Int32 *argnum*) [inline]**

Definition at line 369 of file ChInterp.cs.

References `_Ch_FuncArgArrayDim()`, and `interp`.

**12.23.3.103 Int32 EmbeddedCh::ChInterp::FuncArgArrayExtent (String *funcname*, Int32 *argnum*, Int32 *index*) [inline]**

Definition at line 374 of file ChInterp.cs.

References `_Ch_FuncArgArrayExtent()`, and `interp`.

**12.23.3.104 Int32 EmbeddedCh::ChInterp::FuncArgArrayNum (String *funcname*, Int32 *argnum*) [inline]**

Definition at line 379 of file ChInterp.cs.

References `_Ch_FuncArgArrayNum()`, and `interp`.

**12.23.3.105 ChType\_t EmbeddedCh::ChInterp::FuncArgArrayType (String *funcname*, Int32 *argnum*) [inline]**

Definition at line 364 of file ChInterp.cs.

References `_Ch_FuncArgArrayType()`, and `interp`.

**12.23.3.106 ChType\_t EmbeddedCh::ChInterp::FuncArgDataType (String *funcname*, Int32 *argnum*) [inline]**

Definition at line 359 of file ChInterp.cs.

References `_Ch_FuncArgDataType()`, and `interp`.

**12.23.3.107 Int32 EmbeddedCh::ChInterp::FuncArgFuncArgNum (String *funcname*, Int32 *argnum*) [inline]**

Definition at line 394 of file ChInterp.cs.

References `_Ch_FuncArgFuncArgNum()`, and `interp`.

**12.23.3.108 Int32 EmbeddedCh::ChInterp::FuncArgIsFunc (String *funcname*, Int32 *argnum*) [inline]**

Definition at line 384 of file ChInterp.cs.

References `_Ch_FuncArgIsFunc()`, and `interp`.

**12.23.3.109 Int32 EmbeddedCh::ChInterp::FuncArgIsFuncVarArg (String *funcname*, Int32 *argnum*) [inline]**

Definition at line 389 of file ChInterp.cs.

References `_Ch_FuncArgIsFuncVarArg()`, and `interp`.

**12.23.3.110 Int32 EmbeddedCh::ChInterp::FuncArgNum (String *name*) [inline]**

Definition at line 299 of file ChInterp.cs.

References `_Ch_FuncArgNum()`, and `interp`.

**12.23.3.111 String EmbeddedCh::ChInterp::FuncArgUserDefinedName (String *funcname*, Int32 *argnum*) [inline]**

Definition at line 399 of file ChInterp.cs.

References `_Ch_FuncArgUserDefinedName()`, and `interp`.

**12.23.3.112 Int32 EmbeddedCh::ChInterp::FuncArgUserDefinedSize (String *funcname*, Int32 *argnum*) [inline]**

Definition at line 404 of file ChInterp.cs.

References `_Ch_FuncArgUserDefinedSize()`, and `interp`.

**12.23.3.113 ChFuncType\_t EmbeddedCh::ChInterp::FuncType (String *name*) [inline]**

Definition at line 289 of file ChInterp.cs.

References `_Ch_FuncType()`, and `interp`.

**12.23.3.114 IntPtr EmbeddedCh::ChInterp::GetGlobalUserData () [inline]**

Definition at line 208 of file ChInterp.cs.

References `_Ch_GetGlobalUserData()`, and `interp`.

**12.23.3.115   IntPtr EmbeddedCh::ChInterp::GlobalSymbolAddrByIndex (Int32 *index*)  
[inline]**

Definition at line 244 of file ChInterp.cs.

References `_Ch_GlobalSymbolAddrByIndex()`, and `interp`.

**12.23.3.116   Int32 EmbeddedCh::ChInterp::GlobalSymbolIndexByName (String *name*)  
[inline]**

Definition at line 239 of file ChInterp.cs.

References `_Ch_GlobalSymbolIndexByName()`, and `interp`.

**12.23.3.117   String EmbeddedCh::ChInterp::GlobalSymbolNameByIndex (Int32 *index*)  
[inline]**

Definition at line 249 of file ChInterp.cs.

References `_Ch_GlobalSymbolNameByIndex()`, and `interp`.

**12.23.3.118   Int32 EmbeddedCh::ChInterp::GlobalSymbolTotalNum ()   [inline]**

Definition at line 234 of file ChInterp.cs.

References `_Ch_GlobalSymbolTotalNum()`, and `interp`.

**12.23.3.119   int EmbeddedCh::ChInterp::InitGlobalVar (Int32 *flag*)   [inline]**

Definition at line 88 of file ChInterp.cs.

References `_Ch_InitGlobalVar()`, and `interp`.

**12.23.3.120   int EmbeddedCh::ChInterp::Initialize ()   [inline]**

Definition at line 65 of file ChInterp.cs.

References `_Ch_Initialize()`, `interp`, and `useOptions`.

Referenced by `Program1::Program::Main()`.

**12.23.3.121   Int32 EmbeddedCh::ChInterp::IsFuncVarArg (String *name*)   [inline]**

Definition at line 294 of file ChInterp.cs.

References `_Ch_IsFuncVarArg()`, and `interp`.

**12.23.3.122   int EmbeddedCh::ChInterp::ParseScript (String[] *argv*)   [inline]**

Definition at line 93 of file ChInterp.cs.

References `_Ch_ParseScript()`, and `interp`.

**12.23.3.123 Int32 EmbeddedCh::ChInterp::Reopen (String *filename*, String *mode*, Int32 *filde*) [inline]**

Definition at line 173 of file ChInterp.cs.

References `_Ch_Reopen()`, and `interp`.

**12.23.3.124 int EmbeddedCh::ChInterp::RunScript (String[ ] *argv*) [inline]**

Definition at line 73 of file ChInterp.cs.

References `_Ch_RunScript()`, and `interp`.

Referenced by `Program1::Program::Main()`.

**12.23.3.125 int EmbeddedCh::ChInterp::RunScriptM (String[ ] *argv*) [inline]**

Definition at line 108 of file ChInterp.cs.

References `_Ch_RunScriptM()`, and `interp`.

**12.23.3.126 Int32 EmbeddedCh::ChInterp::SetGlobalUserData (IntPtr *userdata*) [inline]**

Definition at line 203 of file ChInterp.cs.

References `_Ch_SetGlobalUserData()`, and `interp`.

**12.23.3.127 Int32 EmbeddedCh::ChInterp::SetVar (String *name*, ChType\_t *atype*) [inline]**

Definition at line 159 of file ChInterp.cs.

References `_Ch_SetVar()`, and `interp`.

**12.23.3.128 Int32 EmbeddedCh::ChInterp::StackLevel (ref Int32 *clevel*, ref Int32 *hlevel*) [inline]**

Definition at line 224 of file ChInterp.cs.

References `_Ch_StackLevel()`, and `interp`.

**12.23.3.129 String EmbeddedCh::ChInterp::StackName (Int32 *level*, ref Int32 *isfunc*, String[ ] *classname*) [inline]**

Definition at line 229 of file ChInterp.cs.

References `_Ch_StackName()`, and `interp`.

**12.23.3.130 IntPtr EmbeddedCh::ChInterp::SymbolAddrByIndex (Int32 *index*) [inline]**

Definition at line 319 of file ChInterp.cs.

References `_Ch_SymbolAddrByIndex()`, and `interp`.

**12.23.3.131 IntPtr EmbeddedCh::ChInterp::SymbolAddrByName (String *name*) [inline]**

Definition at line 304 of file ChInterp.cs.

References `_Ch_SymbolAddrByName()`, and `interp`.

**12.23.3.132 Int32 EmbeddedCh::ChInterp::SymbolIndexByName (String *name*) [inline]**

Definition at line 314 of file ChInterp.cs.

References `_Ch_SymbolIndexByName()`, and `interp`.

**12.23.3.133 String EmbeddedCh::ChInterp::SymbolNameByIndex (Int32 *index*) [inline]**

Definition at line 324 of file ChInterp.cs.

References `_Ch_SymbolNameByIndex()`, and `interp`.

**12.23.3.134 Int32 EmbeddedCh::ChInterp::SymbolTotalNum () [inline]**

Definition at line 309 of file ChInterp.cs.

References `_Ch_SymbolTotalNum()`, and `interp`.

**12.23.3.135 Int32 EmbeddedCh::ChInterp::UserDefinedInfo (IntPtr *udtag*, ref ChUserDefinedInfo\_t *udinfo*) [inline]**

Definition at line 334 of file ChInterp.cs.

References `_Ch_UserDefinedInfo()`, and `interp`.

**12.23.3.136 Int32 EmbeddedCh::ChInterp::UserDefinedMemInfoByIndex (IntPtr *udtag*, Int32 *index*, ref ChMemInfo\_t *meminfo*) [inline]**

Definition at line 344 of file ChInterp.cs.

References `_Ch_UserDefinedMemInfoByIndex()`, and `interp`.

**12.23.3.137 Int32 EmbeddedCh::ChInterp::UserDefinedMemInfoByName (IntPtr *udtag*, String *memname*, ref ChMemInfo\_t *meminfo*) [inline]**

Definition at line 339 of file ChInterp.cs.

References `_Ch_UserDefinedMemInfoByName()`, and `interp`.

**12.23.3.138 String EmbeddedCh::ChInterp::UserDefinedName (String *name*) [inline]**

Definition at line 349 of file ChInterp.cs.

References `_Ch_UserDefinedName()`, and `interp`.

**12.23.3.139 Int32 EmbeddedCh::ChInterp::UserDefinedSize (String *name*) [inline]**

Definition at line 354 of file ChInterp.cs.

References `_Ch_UserDefinedSize()`, and `interp`.

**12.23.3.140 IntPtr EmbeddedCh::ChInterp::UserDefinedTag (String *expr*) [inline]**

Definition at line 329 of file ChInterp.cs.

References `_Ch_UserDefinedTag()`, and `interp`.

**12.23.3.141 ChVarType\_t EmbeddedCh::ChInterp::VarType (String *name*) [inline]**

Definition at line 284 of file ChInterp.cs.

References `_Ch_VarType()`, and `interp`.

**12.23.4 Field Documentation****12.23.4.1 const String EmbeddedCh::ChInterp::chdll = "embedchdll.dll"**

Definition at line 13 of file ChInterp.cs.

**12.23.4.2 IntPtr EmbeddedCh::ChInterp::interp = IntPtr.Zero [private]**

Definition at line 19 of file ChInterp.cs.

Referenced by `Abort()`, `AddCallback()`, `AppendParseScript()`, `AppendParseScriptFile()`, `AppendRunScript()`, `AppendRunScriptFile()`, `ArrayDim()`, `ArrayExtent()`, `ArrayNum()`, `ArrayType()`, `ChangeStack()`, `Close()`, `DataSize()`, `DataType()`, `DeclareFunc()`, `DeclareTypedef()`, `DeclareVar()`, `DeleteExprValue()`, `End()`, `ExecScript()`, `ExecScriptM()`, `ExprCalc()`, `ExprEval()`, `ExprParse()`, `ExprValue()`, `Flush()`, `FuncArgArrayDim()`, `FuncArgArrayExtent()`, `FuncArgArrayNum()`, `FuncArgArrayType()`, `FuncArgDataType()`, `FuncArgFuncArgNum()`, `FuncArgIsFunc()`, `FuncArgIsFuncVarArg()`, `FuncArgNum()`, `FuncArgUserDefinedName()`, `FuncArgUserDefinedSize()`, `FuncType()`, `GetGlobalUserData()`, `GlobalSymbolAddrByIndex()`, `GlobalSymbolIndexByName()`, `GlobalSymbolNameByIndex()`, `GlobalSymbolTotalNum()`, `InitGlobalVar()`, `Initialize()`, `IsFuncVarArg()`, `ParseScript()`, `Reopen()`, `RunScript()`, `RunScriptM()`, `SetGlobalUserData()`, `SetVar()`, `StackLevel()`, `StackName()`, `SymbolAddrByIndex()`, `SymbolAddrByName()`, `SymbolIndexByName()`, `SymbolNameByIndex()`, `SymbolTotalNum()`, `UserDefinedInfo()`, `UserDefinedMemInfoByIndex()`, `UserDefinedMemInfoByName()`, `UserDefinedName()`, `UserDefinedSize()`, `UserDefinedTag()`, and `VarType()`.

**12.23.4.3 ChOptions\_t EmbeddedCh::ChInterp::options = new ChOptions\_t() [private]**

Definition at line 22 of file ChInterp.cs.

**12.23.4.4 bool EmbeddedCh::ChInterp::useOptions = false [private]**

Definition at line 23 of file ChInterp.cs.

Referenced by `Initialize()`.

## 12.23.5 Property Documentation

### 12.23.5.1 String EmbeddedCh::ChInterp::ChHome [get, set]

Definition at line 54 of file ChInterp.cs.

### 12.23.5.2 ChShellType EmbeddedCh::ChInterp::ShellType [get, set]

Definition at line 42 of file ChInterp.cs.

### 12.23.5.3 bool EmbeddedCh::ChInterp::UseOptions [get, set]

Definition at line 30 of file ChInterp.cs.

The documentation for this class was generated from the following file:

- /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/win32/EmbeddedCh.Net/EmbeddedCh.Net/[ChInterp.cs](#)



## 12.24 EmbeddedCh::ChMemInfo\_t Struct Reference

### Data Fields

- [int](#) `index`
- [String](#) `memname`
- [int](#) `offset`
- [ChType\\_t](#) `dtype`
- [int](#) `ispublic`
- [int](#) `isfunc`
- [int](#) `ismemberfunc`
- [int](#) `isconstructor`
- [int](#) `isdestructor`
- [int](#) `isvararg`
- [int](#) `arraytype`
- [int](#) `dim`
- [int\[ \]](#) `extent`
- [int](#) `isbitfield`
- [int](#) `fieldsize`
- [int](#) `fieldoffset`
- [IntPtr](#) `udtag`

### 12.24.1 Detailed Description

Definition at line 72 of file EmbedCh.cs.

### 12.24.2 Field Documentation

#### 12.24.2.1 `int EmbeddedCh::ChMemInfo_t::arraytype`

Definition at line 83 of file EmbedCh.cs.

#### 12.24.2.2 `int EmbeddedCh::ChMemInfo_t::dim`

Definition at line 84 of file EmbedCh.cs.

#### 12.24.2.3 `ChType_t EmbeddedCh::ChMemInfo_t::dtype`

Definition at line 76 of file EmbedCh.cs.

#### 12.24.2.4 `int [ ] EmbeddedCh::ChMemInfo_t::extent`

Definition at line 86 of file EmbedCh.cs.

#### 12.24.2.5 `int EmbeddedCh::ChMemInfo_t::fieldoffset`

Definition at line 89 of file EmbedCh.cs.

**12.24.2.6 int EmbeddedCh::ChMemInfo\_t::fieldsize**

Definition at line 88 of file EmbedCh.cs.

**12.24.2.7 int EmbeddedCh::ChMemInfo\_t::index**

Definition at line 73 of file EmbedCh.cs.

**12.24.2.8 int EmbeddedCh::ChMemInfo\_t::isbitfield**

Definition at line 87 of file EmbedCh.cs.

**12.24.2.9 int EmbeddedCh::ChMemInfo\_t::isconstructor**

Definition at line 80 of file EmbedCh.cs.

**12.24.2.10 int EmbeddedCh::ChMemInfo\_t::isdestructor**

Definition at line 81 of file EmbedCh.cs.

**12.24.2.11 int EmbeddedCh::ChMemInfo\_t::isfunc**

Definition at line 78 of file EmbedCh.cs.

**12.24.2.12 int EmbeddedCh::ChMemInfo\_t::ismemberfunc**

Definition at line 79 of file EmbedCh.cs.

**12.24.2.13 int EmbeddedCh::ChMemInfo\_t::ispublic**

Definition at line 77 of file EmbedCh.cs.

**12.24.2.14 int EmbeddedCh::ChMemInfo\_t::isvararg**

Definition at line 82 of file EmbedCh.cs.

**12.24.2.15 String EmbeddedCh::ChMemInfo\_t::memname**

Definition at line 74 of file EmbedCh.cs.

**12.24.2.16 int EmbeddedCh::ChMemInfo\_t::offset**

Definition at line 75 of file EmbedCh.cs.

**12.24.2.17 IntPtr EmbeddedCh::ChMemInfo\_t::udtag**

Definition at line 92 of file EmbedCh.cs.

The documentation for this struct was generated from the following file:

- </home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/win32/EmbeddedCh.Net/EmbeddedCh.Net/EmbedCh.cs>

## 12.25 EmbeddedCh::ChOptions\_t Struct Reference

### Data Fields

- [int shelltype](#)
- [String chhome](#)

### 12.25.1 Detailed Description

Definition at line 9 of file EmbedCh.cs.

### 12.25.2 Field Documentation

#### 12.25.2.1 String EmbeddedCh::ChOptions\_t::chhome

Embedded Ch home directory. if NULL, use Ch home directory for standard/professional edition as Embedded Ch home directory

Definition at line 16 of file EmbedCh.cs.

#### 12.25.2.2 int EmbeddedCh::ChOptions\_t::shelltype

shell type: CH\_REGULARCH or CH\_SAFECH

Definition at line 11 of file EmbedCh.cs.

The documentation for this struct was generated from the following file:

- [/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/win32/EmbeddedCh.Net/EmbeddedCh.Net/EmbedCh.cs](#)

## 12.26 LibMC::MCAgency::ChOptions\_t Struct Reference

ChOptions structures.

### Data Fields

- [int shelltype](#)
- [String chhome](#)

### 12.26.1 Detailed Description

ChOptions structures. Allows the user to set the shell type and the home directory.

#### Note:

This struct is pulled directly from the Mobile-C library.

Definition at line 61 of file MCEExports.cs.

### 12.26.2 Field Documentation

#### 12.26.2.1 String LibMC::MCAgency::ChOptions\_t::chhome

Embedded Ch home directory. if NULL, use Ch home directory for standard/professional edition as Embedded Ch home directory

Definition at line 66 of file MCEExports.cs.

Referenced by LibMC::MCAgency::ChInitializeOptions().

#### 12.26.2.2 int LibMC::MCAgency::ChOptions\_t::shelltype

shell type: CH\_REGULARCH or CH\_SAFECH

Definition at line 63 of file MCEExports.cs.

Referenced by LibMC::MCAgency::ChInitializeOptions().

The documentation for this struct was generated from the following file:

- </home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/win32/LibMC.net/LibMC/MCEExports.cs>

## 12.27 EmbeddedCh::ChUserDefinedInfo\_t Struct Reference

### Data Fields

- [ChType\\_t dtype](#)
- [String tagname](#)
- [int size](#)
- [int totnum](#)

### 12.27.1 Detailed Description

Definition at line 64 of file EmbedCh.cs.

### 12.27.2 Field Documentation

#### 12.27.2.1 ChType\_t EmbeddedCh::ChUserDefinedInfo\_t::dtype

Definition at line 65 of file EmbedCh.cs.

#### 12.27.2.2 int EmbeddedCh::ChUserDefinedInfo\_t::size

Definition at line 67 of file EmbedCh.cs.

#### 12.27.2.3 String EmbeddedCh::ChUserDefinedInfo\_t::tagname

Definition at line 66 of file EmbedCh.cs.

#### 12.27.2.4 int EmbeddedCh::ChUserDefinedInfo\_t::totnum

Definition at line 68 of file EmbedCh.cs.

The documentation for this struct was generated from the following file:

- [/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/win32/EmbeddedCh.Net/EmbeddedCh.Net/EmbedCh.cs](#)

## 12.28 EmbeddedCh::ChUserDefinedTag Class Reference

### Public Member Functions

- [ChUserDefinedTag \(\)](#)

### Properties

- [IntPtr Tag](#) [get, set]

### Private Attributes

- [IntPtr ptr](#)

#### 12.28.1 Detailed Description

Definition at line 8 of file ChUserDefinedTag.cs.

#### 12.28.2 Constructor & Destructor Documentation

##### 12.28.2.1 EmbeddedCh::ChUserDefinedTag::ChUserDefinedTag () [inline]

Definition at line 12 of file ChUserDefinedTag.cs.

References [ptr](#).

#### 12.28.3 Field Documentation

##### 12.28.3.1 IntPtr EmbeddedCh::ChUserDefinedTag::ptr [private]

Definition at line 10 of file ChUserDefinedTag.cs.

Referenced by [ChUserDefinedTag\(\)](#).

#### 12.28.4 Property Documentation

##### 12.28.4.1 IntPtr EmbeddedCh::ChUserDefinedTag::Tag [get, set]

Definition at line 18 of file ChUserDefinedTag.cs.

The documentation for this class was generated from the following file:

- [/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/win32/EmbeddedCh.Net/EmbeddedCh.Net/ChUserDefinedTag.](#)

## 12.29 EmbeddedCh::ChVaList Class Reference

### Public Member Functions

- [ChVaList \(\)](#)

### Private Attributes

- IntPtr [ptr](#) = IntPtr.Zero

#### 12.29.1 Detailed Description

Definition at line 8 of file ChVaList.cs.

#### 12.29.2 Constructor & Destructor Documentation

##### 12.29.2.1 EmbeddedCh::ChVaList::ChVaList () [inline]

Definition at line 12 of file ChVaList.cs.

#### 12.29.3 Field Documentation

##### 12.29.3.1 IntPtr EmbeddedCh::ChVaList::ptr = IntPtr.Zero [private]

Definition at line 10 of file ChVaList.cs.

The documentation for this class was generated from the following file:

- /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/win32/EmbeddedCh.Net/EmbeddedCh.Net/[ChVaList.cs](#)



## 12.30 cmd\_prompt\_s Struct Reference

```
#include <cmd_prompt.h>
```

### Data Fields

- `THREAD_T` [thread](#)

### 12.30.1 Detailed Description

Definition at line 41 of file `cmd_prompt.h`.

### 12.30.2 Field Documentation

#### 12.30.2.1 `THREAD_T cmd_prompt_s::thread`

Definition at line 42 of file `cmd_prompt.h`.

Referenced by `cmd_prompt_Start()`, and `MC_End()`.

The documentation for this struct was generated from the following file:

- `/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/cmd\_prompt.h`

## 12.31 `command_s` Struct Reference

```
#include <cmd_prompt.h>
```

### Data Fields

- [int index](#)
- [int num\\_args](#)
- [char \\*\\* args](#)

### 12.31.1 Detailed Description

Definition at line 46 of file `cmd_prompt.h`.

### 12.31.2 Field Documentation

#### 12.31.2.1 `char** command_s::args`

Definition at line 49 of file `cmd_prompt.h`.

Referenced by `cmd_prompt_Thread()`, `dealloc_command()`, `exec_command()`, `handler_HELP()`, `handler_SEND()`, and `process_command()`.

#### 12.31.2.2 `int command_s::index`

Definition at line 47 of file `cmd_prompt.h`.

Referenced by `cmd_prompt_Thread()`, `exec_command()`, and `process_command()`.

#### 12.31.2.3 `int command_s::num_args`

Definition at line 48 of file `cmd_prompt.h`.

Referenced by `cmd_prompt_Thread()`, `dealloc_command()`, `exec_command()`, `handler_HELP()`, `handler_SEND()`, and `process_command()`.

The documentation for this struct was generated from the following file:

- [/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/cmd\\_prompt.h](#)

## 12.32 connection\_s Struct Reference

```
#include <connection.h>
```

### Data Fields

- [int connect\\_id](#)
- [int nonce](#)
- [char \\* remote\\_hostname](#)
- [struct sockaddr\\_in addr](#)
- [u\\_long clientfd](#)
- [u\\_long serverfd](#)
- [unsigned char \\* AES\\_key](#)

### 12.32.1 Detailed Description

Definition at line 46 of file connection.h.

### 12.32.2 Field Documentation

#### 12.32.2.1 struct sockaddr\_in connection\_s::addr [read]

Definition at line 56 of file connection.h.

Referenced by connection\_Copy(), listen\_Thread(), and message\_InitializeFromConnection().

#### 12.32.2.2 unsigned char\* connection\_s::AES\_key

Definition at line 63 of file connection.h.

Referenced by listen\_Thread(), and rece\_de\_msg().

#### 12.32.2.3 u\_long connection\_s::clientfd

Definition at line 59 of file connection.h.

Referenced by connection\_Copy(), connection\_Destroy(), listen\_Thread(), message\_InitializeFromConnection(), mtp\_http\_InitializeFromConnection(), and rece\_de\_msg().

#### 12.32.2.4 int connection\_s::connect\_id

Definition at line 49 of file connection.h.

Referenced by AP\_QUEUE\_SEARCH\_TEMPLATE(), connection\_Copy(), listen\_Thread(), and message\_InitializeFromConnection().

#### 12.32.2.5 int connection\_s::nonce

Definition at line 51 of file connection.h.

Referenced by listen\_Thread(), and rece\_de\_msg().

**12.32.2.6 char\* connection\_s::remote\_hostname**

Definition at line 54 of file connection.h.

Referenced by AP\_QUEUE\_SEARCH\_TEMPLATE(), connection\_Copy(), connection\_Destroy(), and listen\_Thread().

**12.32.2.7 u\_long connection\_s::serverfd**

Definition at line 60 of file connection.h.

Referenced by connection\_Copy(), and listen\_Thread().

The documentation for this struct was generated from the following file:

- /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/[connection.h](#)

## 12.33 fipa\_expression\_s::content\_u Union Reference

```
#include <fipa_acl.h>
```

### Data Fields

- struct [fipa\\_word\\_s](#) \* word
- struct [fipa\\_string\\_s](#) \* string
- struct [fipa\\_number\\_s](#) \* number
- struct [fipa\\_DateTime\\_s](#) \* datetime
- struct [fipa\\_expression\\_s](#) \*\* expression

#### 12.33.1 Detailed Description

Definition at line 196 of file [fipa\\_acl.h](#).

#### 12.33.2 Field Documentation

##### 12.33.2.1 struct [fipa\\_DateTime\\_s](#)\* [fipa\\_expression\\_s::content\\_u::datetime](#) [read]

Definition at line 201 of file [fipa\\_acl.h](#).

Referenced by [fipa\\_expression\\_Compose\(\)](#), [fipa\\_expression\\_Copy\(\)](#), and [fipa\\_expression\\_Destroy\(\)](#).

##### 12.33.2.2 struct [fipa\\_expression\\_s](#)\*\* [fipa\\_expression\\_s::content\\_u::expression](#) [read]

Definition at line 202 of file [fipa\\_acl.h](#).

Referenced by [fipa\\_expression\\_Compose\(\)](#), [fipa\\_expression\\_Copy\(\)](#), and [fipa\\_expression\\_Destroy\(\)](#).

##### 12.33.2.3 struct [fipa\\_number\\_s](#)\* [fipa\\_expression\\_s::content\\_u::number](#) [read]

Definition at line 200 of file [fipa\\_acl.h](#).

Referenced by [fipa\\_expression\\_Compose\(\)](#), [fipa\\_expression\\_Copy\(\)](#), and [fipa\\_expression\\_Destroy\(\)](#).

##### 12.33.2.4 struct [fipa\\_string\\_s](#)\* [fipa\\_expression\\_s::content\\_u::string](#) [read]

Definition at line 199 of file [fipa\\_acl.h](#).

Referenced by [fipa\\_expression\\_Compose\(\)](#), [fipa\\_expression\\_Copy\(\)](#), [fipa\\_expression\\_Destroy\(\)](#), and [MC\\_AclSetConversationID\(\)](#).

##### 12.33.2.5 struct [fipa\\_word\\_s](#)\* [fipa\\_expression\\_s::content\\_u::word](#) [read]

Definition at line 198 of file [fipa\\_acl.h](#).

Referenced by [fipa\\_expression\\_Compose\(\)](#), [fipa\\_expression\\_Copy\(\)](#), and [fipa\\_expression\\_Destroy\(\)](#).

The documentation for this union was generated from the following file:

- [/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/fipa\\_acl.h](#)

## 12.34 des3\_context Struct Reference

Triple-DES context structure.

```
#include <des.h>
```

### Data Fields

- [int mode](#)
- unsigned long [sk](#) [96]

### 12.34.1 Detailed Description

Triple-DES context structure.

Definition at line 23 of file des.h.

### 12.34.2 Field Documentation

#### 12.34.2.1 int des3\_context::mode

encrypt/decrypt

Definition at line 25 of file des.h.

#### 12.34.2.2 unsigned long des3\_context::sk[96]

3DES subkeys

Definition at line 26 of file des.h.

Referenced by `des3_crypt_ecb()`, `des3_set2key_dec()`, `des3_set2key_enc()`, `des3_set3key_dec()`, and `des3_set3key_enc()`.

The documentation for this struct was generated from the following file:

- `/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/des.h`

## 12.35 des\_context Struct Reference

DES context structure.

```
#include <des.h>
```

### Data Fields

- [int mode](#)
- unsigned long [sk](#) [32]

### 12.35.1 Detailed Description

DES context structure.

Definition at line 13 of file des.h.

### 12.35.2 Field Documentation

#### 12.35.2.1 int des\_context::mode

encrypt/decrypt

Definition at line 15 of file des.h.

#### 12.35.2.2 unsigned long des\_context::sk[32]

DES subkeys

Definition at line 16 of file des.h.

Referenced by [des\\_crypt\\_ecb\(\)](#), [des\\_setkey\\_dec\(\)](#), and [des\\_setkey\\_enc\(\)](#).

The documentation for this struct was generated from the following file:

- [/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/des.h](#)



## 12.36 dhm\_context Struct Reference

```
#include <dhm.h>
```

### Data Fields

- [int len](#)
- [mpi P](#)
- [mpi G](#)
- [mpi X](#)
- [mpi GX](#)
- [mpi GY](#)
- [mpi K](#)
- [mpi RP](#)

### 12.36.1 Detailed Description

Definition at line 16 of file dhm.h.

### 12.36.2 Field Documentation

#### 12.36.2.1 mpi dhm\_context::G

generator

Definition at line 20 of file dhm.h.

Referenced by dhm\_free(), dhm\_make\_params(), dhm\_make\_public(), dhm\_read\_params(), main(), ssl\_parse\_server\_key\_exchange(), ssl\_set\_dh\_param(), and ssl\_write\_server\_key\_exchange().

#### 12.36.2.2 mpi dhm\_context::GX

$\text{self} = G^X \bmod P$

Definition at line 22 of file dhm.h.

Referenced by dhm\_free(), dhm\_make\_params(), dhm\_make\_public(), ssl\_write\_client\_key\_exchange(), and ssl\_write\_server\_key\_exchange().

#### 12.36.2.3 mpi dhm\_context::GY

$\text{peer} = G^Y \bmod P$

Definition at line 23 of file dhm.h.

Referenced by dhm\_calc\_secret(), dhm\_free(), dhm\_read\_params(), dhm\_read\_public(), ssl\_parse\_client\_key\_exchange(), and ssl\_parse\_server\_key\_exchange().

#### 12.36.2.4 `mpi dhm_context::K`

$\text{key} = \text{GY}^X \bmod P$

Definition at line 24 of file dhm.h.

Referenced by `dhm_calc_secret()`, `dhm_free()`, `ssl_parse_client_key_exchange()`, and `ssl_write_client_key_exchange()`.

#### 12.36.2.5 `int dhm_context::len`

`size(P)` in chars

Definition at line 18 of file dhm.h.

Referenced by `dhm_make_params()`, `dhm_make_public()`, `dhm_read_params()`, `dhm_read_public()`, `main()`, `ssl_parse_client_key_exchange()`, `ssl_parse_server_key_exchange()`, and `ssl_write_client_key_exchange()`.

#### 12.36.2.6 `mpi dhm_context::P`

prime modulus

Definition at line 19 of file dhm.h.

Referenced by `dhm_calc_secret()`, `dhm_free()`, `dhm_make_params()`, `dhm_make_public()`, `dhm_read_params()`, `main()`, `ssl_parse_server_key_exchange()`, `ssl_set_dh_param()`, and `ssl_write_server_key_exchange()`.

#### 12.36.2.7 `mpi dhm_context::RP`

cached  $R^2 \bmod P$

Definition at line 25 of file dhm.h.

Referenced by `dhm_calc_secret()`, `dhm_free()`, `dhm_make_params()`, and `dhm_make_public()`.

#### 12.36.2.8 `mpi dhm_context::X`

secret value

Definition at line 21 of file dhm.h.

Referenced by `dhm_calc_secret()`, `dhm_free()`, `dhm_make_params()`, `dhm_make_public()`, `ssl_write_client_key_exchange()`, and `ssl_write_server_key_exchange()`.

The documentation for this struct was generated from the following file:

- `/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/dhm.h`

## 12.37 dynstring\_s Struct Reference

```
#include <dynstring.h>
```

### Data Fields

- [int len](#)
- [int size](#)
- [char \\* message](#)

### 12.37.1 Detailed Description

Definition at line 9 of file dynstring.h.

### 12.37.2 Field Documentation

#### 12.37.2.1 int dynstring\_s::len

Definition at line 10 of file dynstring.h.

Referenced by [dynstring\\_Append\(\)](#), [dynstring\\_New\(\)](#), and [mtp\\_http\\_CreateMessage\(\)](#).

#### 12.37.2.2 char\* dynstring\_s::message

Definition at line 12 of file dynstring.h.

Referenced by [dynstring\\_Append\(\)](#), [dynstring\\_Destroy\(\)](#), [dynstring\\_New\(\)](#), [fipa\\_envelope\\_Compose\\_\\_\\_from\(\)](#), [MC\\_AclSend\(\)](#), [message\\_send\\_Thread\(\)](#), and [mtp\\_http\\_CreateMessage\(\)](#).

#### 12.37.2.3 int dynstring\_s::size

Definition at line 11 of file dynstring.h.

Referenced by [dynstring\\_Append\(\)](#), and [dynstring\\_New\(\)](#).

The documentation for this struct was generated from the following file:

- [/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/dynstring.h](#)

## 12.38 fipa\_acl\_envelope\_Received\_s Struct Reference

```
#include <fipa_acl_envelope.h>
```

### Data Fields

- struct [fipa\\_url\\_s](#) \* [received\\_by](#)
- struct [fipa\\_url\\_s](#) \* [received\\_from](#)
- struct [fipa\\_DateTime\\_s](#) \* [received\\_date](#)
- char \* [received\\_id](#)
- struct [fipa\\_url\\_s](#) \* [received\\_via](#)

### 12.38.1 Detailed Description

Definition at line 7 of file `fipa_acl_envelope.h`.

### 12.38.2 Field Documentation

#### 12.38.2.1 struct `fipa_url_s`\* `fipa_acl_envelope_Received_s::received_by` [read]

Definition at line 9 of file `fipa_acl_envelope.h`.

Referenced by `fipa_acl_envelope_Received_Copy()`, `fipa_acl_envelope_Received_Destroy()`, and `fipa_envelope_HandleReceived()`.

#### 12.38.2.2 struct `fipa_DateTime_s`\* `fipa_acl_envelope_Received_s::received_date` [read]

Definition at line 11 of file `fipa_acl_envelope.h`.

Referenced by `fipa_acl_envelope_Received_Copy()`, `fipa_acl_envelope_Received_Destroy()`, and `fipa_envelope_HandleReceived()`.

#### 12.38.2.3 struct `fipa_url_s`\* `fipa_acl_envelope_Received_s::received_from` [read]

Definition at line 10 of file `fipa_acl_envelope.h`.

Referenced by `fipa_acl_envelope_Received_Copy()`, `fipa_acl_envelope_Received_Destroy()`, and `fipa_envelope_HandleReceived()`.

#### 12.38.2.4 char\* `fipa_acl_envelope_Received_s::received_id`

Definition at line 12 of file `fipa_acl_envelope.h`.

Referenced by `fipa_acl_envelope_Received_Copy()`, `fipa_acl_envelope_Received_Destroy()`, and `fipa_envelope_HandleReceived()`.

#### 12.38.2.5 struct `fipa_url_s`\* `fipa_acl_envelope_Received_s::received_via` [read]

Definition at line 13 of file `fipa_acl_envelope.h`.

Referenced by `fipa_acl_envelope_Received_Copy()`, `fipa_acl_envelope_Received_Destroy()`, and `fipa_envelope_HandleReceived()`.

The documentation for this struct was generated from the following file:

- `/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/fipa_acl_envelope.h`

## 12.39 fipa\_acl\_envelope\_s Struct Reference

```
#include <fipa_acl_envelope.h>
```

### Data Fields

- [int num\\_params](#)
- [struct fipa\\_acl\\_Param\\_s \\*\\* params](#)

### 12.39.1 Detailed Description

Definition at line 40 of file [fipa\\_acl\\_envelope.h](#).

### 12.39.2 Field Documentation

#### 12.39.2.1 int fipa\_acl\_envelope\_s::num\_params

Definition at line 42 of file [fipa\\_acl\\_envelope.h](#).

Referenced by [acc\\_connection\\_Thread\(\)](#), [fipa\\_acl\\_envelope\\_Copy\(\)](#), [fipa\\_acl\\_envelope\\_Destroy\(\)](#), and [fipa\\_envelope\\_HandleParams\(\)](#).

#### 12.39.2.2 struct fipa\_acl\_Param\_s\*\* fipa\_acl\_envelope\_s::params [read]

Definition at line 43 of file [fipa\\_acl\\_envelope.h](#).

Referenced by [acc\\_connection\\_Thread\(\)](#), [fipa\\_acl\\_envelope\\_Copy\(\)](#), [fipa\\_acl\\_envelope\\_Destroy\(\)](#), [fipa\\_envelope\\_HandleAclRepresentation\(\)](#), [fipa\\_envelope\\_HandleComments\(\)](#), [fipa\\_envelope\\_HandleDate\(\)](#), [fipa\\_envelope\\_HandleFrom\(\)](#), [fipa\\_envelope\\_HandleIntendedReceiver\(\)](#), [fipa\\_envelope\\_HandleParams\(\)](#), [fipa\\_envelope\\_HandlePayloadEncoding\(\)](#), [fipa\\_envelope\\_HandlePayloadLength\(\)](#), [fipa\\_envelope\\_HandleReceived\(\)](#), and [fipa\\_envelope\\_HandleTo\(\)](#).

The documentation for this struct was generated from the following file:

- [/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/fipa\\_acl\\_envelope.h](#)

## 12.40 fipa\_acl\_message\_s Struct Reference

```
#include <fipa_acl.h>
```

### Data Fields

- enum [fipa\\_performative\\_e](#) performative
- struct [fipa\\_agent\\_identifier\\_s](#) \* sender
- struct [fipa\\_agent\\_identifier\\_set\\_s](#) \* receiver
- int receiver\_num
- struct [fipa\\_agent\\_identifier\\_set\\_s](#) \* reply\_to
- struct [fipa\\_string\\_s](#) \* content
- struct [fipa\\_expression\\_s](#) \* language
- struct [fipa\\_expression\\_s](#) \* encoding
- struct [fipa\\_expression\\_s](#) \* ontology
- enum [fipa\\_protocol\\_e](#) protocol
- struct [fipa\\_expression\\_s](#) \* conversation\_id
- struct [fipa\\_expression\\_s](#) \* reply\_with
- struct [fipa\\_expression\\_s](#) \* in\_reply\_to
- struct [fipa\\_DateTime\\_s](#) \* reply\_by

### 12.40.1 Detailed Description

Definition at line 105 of file [fipa\\_acl.h](#).

### 12.40.2 Field Documentation

#### 12.40.2.1 struct [fipa\\_string\\_s](#)\* [fipa\\_acl\\_message\\_s::content](#) [read]

Definition at line 117 of file [fipa\\_acl.h](#).

Referenced by [fipa\\_acl\\_Compose\(\)](#), [fipa\\_acl\\_message\\_Copy\(\)](#), [fipa\\_acl\\_message\\_Destroy\(\)](#), [fipa\\_message\\_parameter\\_Parse\(\)](#), [MC\\_AclSetContent\(\)](#), and [MC\\_AclSetContent\\_chdl\(\)](#).

#### 12.40.2.2 struct [fipa\\_expression\\_s](#)\* [fipa\\_acl\\_message\\_s::conversation\\_id](#) [read]

Definition at line 128 of file [fipa\\_acl.h](#).

Referenced by [fipa\\_acl\\_Compose\(\)](#), [fipa\\_acl\\_message\\_Copy\(\)](#), [fipa\\_acl\\_message\\_Destroy\(\)](#), [fipa\\_message\\_parameter\\_Parse\(\)](#), [fipa\\_Reply\(\)](#), and [MC\\_AclSetConversationID\(\)](#).

#### 12.40.2.3 struct [fipa\\_expression\\_s](#)\* [fipa\\_acl\\_message\\_s::encoding](#) [read]

Definition at line 121 of file [fipa\\_acl.h](#).

Referenced by [fipa\\_acl\\_Compose\(\)](#), [fipa\\_acl\\_message\\_Copy\(\)](#), [fipa\\_acl\\_message\\_Destroy\(\)](#), and [fipa\\_message\\_parameter\\_Parse\(\)](#).

**12.40.2.4 struct fipa\_expression\_s\* fipa\_acl\_message\_s::in\_reply\_to [read]**

Definition at line 132 of file fipa\_acl.h.

Referenced by fipa\_acl\_Compose(), fipa\_acl\_message\_Copy(), fipa\_acl\_message\_Destroy(), and fipa\_message\_parameter\_Parse().

**12.40.2.5 struct fipa\_expression\_s\* fipa\_acl\_message\_s::language [read]**

Definition at line 119 of file fipa\_acl.h.

Referenced by fipa\_acl\_Compose(), fipa\_acl\_message\_Copy(), fipa\_acl\_message\_Destroy(), and fipa\_message\_parameter\_Parse().

**12.40.2.6 struct fipa\_expression\_s\* fipa\_acl\_message\_s::ontology [read]**

Definition at line 123 of file fipa\_acl.h.

Referenced by fipa\_acl\_Compose(), fipa\_acl\_message\_Copy(), fipa\_acl\_message\_Destroy(), and fipa\_message\_parameter\_Parse().

**12.40.2.7 enum fipa\_performative\_e fipa\_acl\_message\_s::performative**

Definition at line 107 of file fipa\_acl.h.

Referenced by fipa\_acl\_Compose(), fipa\_acl\_message\_Copy(), fipa\_acl\_Parse(), MC\_AclSetPerformative(), and MC\_AclSetPerformative\_chdl().

**12.40.2.8 enum fipa\_protocol\_e fipa\_acl\_message\_s::protocol**

Definition at line 126 of file fipa\_acl.h.

Referenced by fipa\_acl\_Compose(), fipa\_acl\_message\_Copy(), fipa\_message\_parameter\_Parse(), fipa\_Reply(), MC\_AclSetProtocol(), and MC\_AclSetProtocol\_chdl().

**12.40.2.9 struct fipa\_agent\_identifier\_set\_s\* fipa\_acl\_message\_s::receiver [read]**

Definition at line 111 of file fipa\_acl.h.

Referenced by AP\_QUEUE\_STD\_DEFN\_TEMPLATE(), fipa\_acl\_Compose(), fipa\_acl\_message\_Copy(), fipa\_acl\_message\_Destroy(), fipa\_envelope\_Compose\_\_intended\_receiver(), fipa\_envelope\_Compose\_\_to(), fipa\_message\_parameter\_Parse(), fipa\_Reply(), MC\_AclAddReceiver(), and MC\_AclSend().

**12.40.2.10 int fipa\_acl\_message\_s::receiver\_num**

Definition at line 112 of file fipa\_acl.h.

Referenced by MC\_AclAddReceiver().



**12.40.2.11 struct fipa\_DateTime\_s\* fipa\_acl\_message\_s::reply\_by [read]**

Definition at line 134 of file fipa\_acl.h.

Referenced by fipa\_acl\_Compose(), fipa\_acl\_message\_Copy(), fipa\_acl\_message\_Destroy(), and fipa\_message\_parameter\_Parse().

**12.40.2.12 struct fipa\_agent\_identifier\_set\_s\* fipa\_acl\_message\_s::reply\_to [read]**

Definition at line 114 of file fipa\_acl.h.

Referenced by fipa\_acl\_Compose(), fipa\_acl\_message\_Copy(), fipa\_acl\_message\_Destroy(), fipa\_message\_parameter\_Parse(), fipa\_Reply(), and MC\_AclAddReplyTo().

**12.40.2.13 struct fipa\_expression\_s\* fipa\_acl\_message\_s::reply\_with [read]**

Definition at line 130 of file fipa\_acl.h.

Referenced by fipa\_acl\_Compose(), fipa\_acl\_message\_Copy(), fipa\_acl\_message\_Destroy(), and fipa\_message\_parameter\_Parse().

**12.40.2.14 struct fipa\_agent\_identifier\_s\* fipa\_acl\_message\_s::sender [read]**

Definition at line 109 of file fipa\_acl.h.

Referenced by fipa\_acl\_Compose(), fipa\_acl\_message\_Copy(), fipa\_acl\_message\_Destroy(), fipa\_envelope\_Compose\_\_from(), fipa\_message\_parameter\_Parse(), fipa\_Reply(), and MC\_AclSetSender().

The documentation for this struct was generated from the following file:

- [/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/fipa\\_acl.h](#)

## 12.41 fipa\_acl\_Param\_s Struct Reference

```
#include <fipa_acl_envelope.h>
```

### Data Fields

- struct [fipa\\_agent\\_identifier\\_set\\_s](#) \* to
- struct [fipa\\_agent\\_identifier\\_s](#) \* from
- char \* [comments](#)
- char \* [acl\\_representation](#)
- char \* [payload\\_length](#)
- char \* [payload\\_encoding](#)
- struct [fipa\\_DateTime\\_s](#) \* date
- struct [fipa\\_agent\\_identifier\\_set\\_s](#) \* intended\_receiver
- struct [fipa\\_acl\\_envelope\\_Received\\_s](#) \* received

### 12.41.1 Detailed Description

Definition at line 22 of file `fipa_acl_envelope.h`.

### 12.41.2 Field Documentation

#### 12.41.2.1 char\* fipa\_acl\_Param\_s::acl\_representation

Definition at line 27 of file `fipa_acl_envelope.h`.

Referenced by `fipa_acl_Param_Copy()`, `fipa_acl_Param_Destroy()`, and `fipa_envelope_HandleAclRepresentation()`.

#### 12.41.2.2 char\* fipa\_acl\_Param\_s::comments

Definition at line 26 of file `fipa_acl_envelope.h`.

Referenced by `fipa_acl_Param_Copy()`, `fipa_acl_Param_Destroy()`, and `fipa_envelope_HandleComments()`.

#### 12.41.2.3 struct fipa\_DateTime\_s\* fipa\_acl\_Param\_s::date [read]

Definition at line 30 of file `fipa_acl_envelope.h`.

Referenced by `fipa_acl_Param_Copy()`, `fipa_acl_Param_Destroy()`, and `fipa_envelope_HandleDate()`.

#### 12.41.2.4 struct fipa\_agent\_identifier\_s\* fipa\_acl\_Param\_s::from [read]

Definition at line 25 of file `fipa_acl_envelope.h`.

Referenced by `fipa_acl_Param_Copy()`, `fipa_acl_Param_Destroy()`, and `fipa_envelope_HandleFrom()`.

**12.41.2.5 struct fipa\_agent\_identifier\_set\_s\* fipa\_acl\_Param\_s::intended\_receiver [read]**

Definition at line 31 of file fipa\_acl\_envelope.h.

Referenced by fipa\_acl\_Param\_Copy(), fipa\_acl\_Param\_Destroy(), and fipa\_envelope\_HandleIntendedReceiver().

**12.41.2.6 char\* fipa\_acl\_Param\_s::payload\_encoding**

Definition at line 29 of file fipa\_acl\_envelope.h.

Referenced by fipa\_acl\_Param\_Copy(), fipa\_acl\_Param\_Destroy(), and fipa\_envelope\_HandlePayloadEncoding().

**12.41.2.7 char\* fipa\_acl\_Param\_s::payload\_length**

Definition at line 28 of file fipa\_acl\_envelope.h.

Referenced by fipa\_acl\_Param\_Copy(), fipa\_acl\_Param\_Destroy(), and fipa\_envelope\_HandlePayloadLength().

**12.41.2.8 struct fipa\_acl\_envelope\_Received\_s\* fipa\_acl\_Param\_s::received [read]**

Definition at line 32 of file fipa\_acl\_envelope.h.

Referenced by fipa\_acl\_Param\_Copy(), fipa\_acl\_Param\_Destroy(), and fipa\_envelope\_HandleReceived().

**12.41.2.9 struct fipa\_agent\_identifier\_set\_s\* fipa\_acl\_Param\_s::to [read]**

Definition at line 24 of file fipa\_acl\_envelope.h.

Referenced by acc\_connection\_Thread(), fipa\_acl\_Param\_Copy(), fipa\_acl\_Param\_Destroy(), and fipa\_envelope\_HandleTo().

The documentation for this struct was generated from the following file:

- [/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/fipa\\_acl\\_envelope.h](#)

## 12.42 fipa\_agent\_identifier\_s Struct Reference

```
#include <fipa_acl.h>
```

### Data Fields

- char \* [name](#)
- struct [fipa\\_url\\_sequence\\_s](#) \* [addresses](#)
- struct [fipa\\_agent\\_identifier\\_set\\_s](#) \* [resolvers](#)

### 12.42.1 Detailed Description

Definition at line 180 of file [fipa\\_acl.h](#).

### 12.42.2 Field Documentation

#### 12.42.2.1 struct [fipa\\_url\\_sequence\\_s](#)\* [fipa\\_agent\\_identifier\\_s::addresses](#) [read]

Definition at line 183 of file [fipa\\_acl.h](#).

Referenced by [acc\\_connection\\_Thread\(\)](#), [fipa\\_agent\\_identifier\\_Compose\(\)](#), [fipa\\_agent\\_identifier\\_Copy\(\)](#), [fipa\\_agent\\_identifier\\_Destroy\(\)](#), [fipa\\_envelope\\_Compose\\_\\_from\(\)](#), [fipa\\_envelope\\_Compose\\_\\_intended\\_receiver\(\)](#), [fipa\\_envelope\\_Compose\\_\\_to\(\)](#), [fipa\\_envelope\\_ParseAddresses\(\)](#), [MC\\_AclAddReceiver\(\)](#), [MC\\_AclAddReplyTo\(\)](#), [MC\\_AclSend\(\)](#), and [MC\\_AclSetSender\(\)](#).

#### 12.42.2.2 char\* [fipa\\_agent\\_identifier\\_s::name](#)

Definition at line 182 of file [fipa\\_acl.h](#).

Referenced by [acc\\_connection\\_Thread\(\)](#), [AP\\_QUEUE\\_STD\\_DEFN\\_TEMPLATE\(\)](#), [fipa\\_agent\\_identifier\\_Compose\(\)](#), [fipa\\_agent\\_identifier\\_Copy\(\)](#), [fipa\\_agent\\_identifier\\_Destroy\(\)](#), [fipa\\_envelope\\_Compose\\_\\_from\(\)](#), [fipa\\_envelope\\_Compose\\_\\_intended\\_receiver\(\)](#), [fipa\\_envelope\\_Compose\\_\\_to\(\)](#), [MC\\_AclAddReceiver\(\)](#), [MC\\_AclAddReplyTo\(\)](#), [MC\\_AclSend\(\)](#), and [MC\\_AclSetSender\(\)](#).

#### 12.42.2.3 struct [fipa\\_agent\\_identifier\\_set\\_s](#)\* [fipa\\_agent\\_identifier\\_s::resolvers](#) [read]

Definition at line 184 of file [fipa\\_acl.h](#).

Referenced by [fipa\\_agent\\_identifier\\_Compose\(\)](#), [fipa\\_agent\\_identifier\\_Copy\(\)](#), [fipa\\_agent\\_identifier\\_Destroy\(\)](#), and [fipa\\_envelope\\_ParseResolvers\(\)](#).

The documentation for this struct was generated from the following file:

- [/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/fipa\\_acl.h](#)

## 12.43 fipa\_agent\_identifier\_set\_s Struct Reference

```
#include <fipa_acl.h>
```

### Data Fields

- [int num](#)
- [int retain\\_order](#)
- struct [fipa\\_agent\\_identifier\\_s](#) \*\* [fipa\\_agent\\_identifiers](#)

### 12.43.1 Detailed Description

Definition at line 166 of file [fipa\\_acl.h](#).

### 12.43.2 Field Documentation

#### 12.43.2.1 struct [fipa\\_agent\\_identifier\\_s](#)\*\* [fipa\\_agent\\_identifier\\_set\\_s::fipa\\_agent\\_identifiers](#) [read]

Definition at line 170 of file [fipa\\_acl.h](#).

Referenced by [acc\\_connection\\_Thread\(\)](#), [AP\\_QUEUE\\_STD\\_DEFN\\_TEMPLATE\(\)](#), [fipa\\_agent\\_identifier\\_set\\_Compose\(\)](#), [fipa\\_agent\\_identifier\\_set\\_Copy\(\)](#), [fipa\\_agent\\_identifier\\_set\\_Destroy\(\)](#), [fipa\\_envelope\\_Compose\\_\\_intended\\_receiver\(\)](#), [fipa\\_envelope\\_Compose\\_\\_to\(\)](#), [fipa\\_envelope\\_HandleIntendedReceiver\(\)](#), [fipa\\_envelope\\_HandleTo\(\)](#), [fipa\\_envelope\\_ParseResolvers\(\)](#), [fipa\\_Reply\(\)](#), [MC\\_AclAddReceiver\(\)](#), [MC\\_AclAddReplyTo\(\)](#), and [MC\\_AclSend\(\)](#).

#### 12.43.2.2 int [fipa\\_agent\\_identifier\\_set\\_s::num](#)

Definition at line 168 of file [fipa\\_acl.h](#).

Referenced by [acc\\_connection\\_Thread\(\)](#), [fipa\\_agent\\_identifier\\_Compose\(\)](#), [fipa\\_agent\\_identifier\\_set\\_Compose\(\)](#), [fipa\\_agent\\_identifier\\_set\\_Copy\(\)](#), [fipa\\_agent\\_identifier\\_set\\_Destroy\(\)](#), [fipa\\_envelope\\_Compose\\_\\_to\(\)](#), [fipa\\_envelope\\_HandleIntendedReceiver\(\)](#), [fipa\\_envelope\\_HandleTo\(\)](#), [fipa\\_envelope\\_ParseResolvers\(\)](#), [fipa\\_Reply\(\)](#), [MC\\_AclAddReceiver\(\)](#), [MC\\_AclAddReplyTo\(\)](#), and [MC\\_AclSend\(\)](#).

#### 12.43.2.3 int [fipa\\_agent\\_identifier\\_set\\_s::retain\\_order](#)

Definition at line 169 of file [fipa\\_acl.h](#).

Referenced by [fipa\\_agent\\_identifier\\_set\\_Copy\(\)](#), [fipa\\_agent\\_identifier\\_set\\_Parse\(\)](#), [fipa\\_envelope\\_HandleIntendedReceiver\(\)](#), [fipa\\_envelope\\_HandleTo\(\)](#), [fipa\\_envelope\\_ParseResolvers\(\)](#), and [fipa\\_Reply\(\)](#).

The documentation for this struct was generated from the following file:

- [/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/fipa\\_acl.h](#)

## 12.44 fipa\_comm\_action\_t Struct Reference

```
#include <fipa_comm.h>
```

### Data Fields

- enum [fipa\\_comm\\_action\\_type\\_e](#) type
- [fipa\\_comm\\_callback](#) callback
- [fipa\\_comm\\_message\\_check\\_p](#) check\_list
- [fipa\\_comm\\_reply\\_p](#) reply
- struct [fipa\\_comm\\_action\\_t](#) \* next

### 12.44.1 Detailed Description

Definition at line 99 of file [fipa\\_comm.h](#).

### 12.44.2 Field Documentation

#### 12.44.2.1 fipa\_comm\_callback fipa\_comm\_action\_t::callback

Definition at line 102 of file [fipa\\_comm.h](#).

#### 12.44.2.2 fipa\_comm\_message\_check\_p fipa\_comm\_action\_t::check\_list

Definition at line 103 of file [fipa\\_comm.h](#).

#### 12.44.2.3 struct fipa\_comm\_action\_t\* fipa\_comm\_action\_t::next [read]

Definition at line 105 of file [fipa\\_comm.h](#).

#### 12.44.2.4 fipa\_comm\_reply\_p fipa\_comm\_action\_t::reply

Definition at line 104 of file [fipa\\_comm.h](#).

#### 12.44.2.5 enum fipa\_comm\_action\_type\_e fipa\_comm\_action\_t::type

Definition at line 101 of file [fipa\\_comm.h](#).

The documentation for this struct was generated from the following file:

- [/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/fipa\\_comm.h](#)

## 12.45 fipa\_comm\_message\_check\_t Struct Reference

```
#include <fipa_comm.h>
```

### Data Fields

- char \* [macro](#)
- enum [fipa\\_comm\\_check\\_type\\_e](#) type
- struct [fipa\\_comm\\_message\\_check\\_t](#) \* next

### 12.45.1 Detailed Description

Definition at line 72 of file fipa\_comm.h.

### 12.45.2 Field Documentation

#### 12.45.2.1 char\* fipa\_comm\_message\_check\_t::macro

Definition at line 74 of file fipa\_comm.h.

#### 12.45.2.2 struct fipa\_comm\_message\_check\_t\* fipa\_comm\_message\_check\_t::next [read]

Definition at line 76 of file fipa\_comm.h.

#### 12.45.2.3 enum fipa\_comm\_check\_type\_e fipa\_comm\_message\_check\_t::type

Definition at line 75 of file fipa\_comm.h.

The documentation for this struct was generated from the following file:

- /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/[fipa\\_comm.h](#)

## 12.46 fipa\_comm\_performative\_t Struct Reference

```
#include <fipa_comm.h>
```

### Data Fields

- [fipa\\_comm\\_action\\_p action\\_list](#)
- [fipa\\_comm\\_reply\\_p default\\_reply](#)

### 12.46.1 Detailed Description

Definition at line 115 of file `fipa_comm.h`.

### 12.46.2 Field Documentation

#### 12.46.2.1 fipa\_comm\_action\_p fipa\_comm\_performative\_t::action\_list

Definition at line 117 of file `fipa_comm.h`.

#### 12.46.2.2 fipa\_comm\_reply\_p fipa\_comm\_performative\_t::default\_reply

Definition at line 118 of file `fipa_comm.h`.

The documentation for this struct was generated from the following file:

- [/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/fipa\\_comm.h](#)



## 12.47 fipa\_comm\_protocol\_cn\_t Struct Reference

```
#include <fipa_comm.h>
```

### Data Fields

- [protocol\\_contract\\_net\\_callback](#) [contract\\_func](#)
- [fipa\\_comm\\_callback](#) [proposal\\_func](#)
- [char](#) [require\\_inform](#)
- [int](#) [time\\_out](#)
- [char \\*](#) [cfp\\_request](#)
- [fipa\\_list\\_p](#) [replies](#)
- [fipa\\_list\\_p](#) [queue](#)

### 12.47.1 Detailed Description

Definition at line 143 of file `fipa_comm.h`.

### 12.47.2 Field Documentation

#### 12.47.2.1 `char* fipa_comm_protocol_cn_t::cfp_request`

Definition at line 149 of file `fipa_comm.h`.

#### 12.47.2.2 `protocol_contract_net_callback fipa_comm_protocol_cn_t::contract_func`

Definition at line 145 of file `fipa_comm.h`.

#### 12.47.2.3 `fipa_comm_callback fipa_comm_protocol_cn_t::proposal_func`

Definition at line 146 of file `fipa_comm.h`.

#### 12.47.2.4 `fipa_list_p fipa_comm_protocol_cn_t::queue`

Definition at line 153 of file `fipa_comm.h`.

#### 12.47.2.5 `fipa_list_p fipa_comm_protocol_cn_t::replies`

Definition at line 152 of file `fipa_comm.h`.

#### 12.47.2.6 `char fipa_comm_protocol_cn_t::require_inform`

Definition at line 147 of file `fipa_comm.h`.

**12.47.2.7 int fipa\_comm\_protocol\_cn\_t::time\_out**

Definition at line 148 of file fipa\_comm.h.

The documentation for this struct was generated from the following file:

- [/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/fipa\\_comm.h](/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/fipa_comm.h)

## 12.48 fipa\_comm\_protocol\_t Struct Reference

```
#include <fipa_comm.h>
```

### Data Fields

- char [error](#)
- char [state](#)
- char \* [conversation\\_id](#)
- char \* [network](#)
- enum [fipa\\_protocol\\_e](#) type
- void \* [info](#)
- struct [fipa\\_comm\\_protocol\\_t](#) \* [next](#)

### 12.48.1 Detailed Description

Definition at line 163 of file [fipa\\_comm.h](#).

### 12.48.2 Field Documentation

#### 12.48.2.1 char\* fipa\_comm\_protocol\_t::conversation\_id

Definition at line 166 of file [fipa\\_comm.h](#).

#### 12.48.2.2 char fipa\_comm\_protocol\_t::error

Definition at line 164 of file [fipa\\_comm.h](#).

#### 12.48.2.3 void\* fipa\_comm\_protocol\_t::info

Definition at line 169 of file [fipa\\_comm.h](#).

#### 12.48.2.4 char\* fipa\_comm\_protocol\_t::network

Definition at line 167 of file [fipa\\_comm.h](#).

#### 12.48.2.5 struct fipa\_comm\_protocol\_t\* fipa\_comm\_protocol\_t::next [\[read\]](#)

Definition at line 170 of file [fipa\\_comm.h](#).

#### 12.48.2.6 char fipa\_comm\_protocol\_t::state

Definition at line 165 of file [fipa\\_comm.h](#).

**12.48.2.7 enum fipa\_protocol\_e fipa\_comm\_protocol\_t::type**

Definition at line 168 of file fipa\_comm.h.

The documentation for this struct was generated from the following file:

- [/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/fipa\\_comm.h](/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/fipa_comm.h)

## 12.49 fipa\_comm\_reply\_t Struct Reference

```
#include <fipa_comm.h>
```

### Data Fields

- enum [fipa\\_performative\\_e](#) type
- char \* [content](#)

### 12.49.1 Detailed Description

Definition at line 86 of file [fipa\\_comm.h](#).

### 12.49.2 Field Documentation

#### 12.49.2.1 char\* [fipa\\_comm\\_reply\\_t::content](#)

Definition at line 89 of file [fipa\\_comm.h](#).

#### 12.49.2.2 enum [fipa\\_performative\\_e](#) [fipa\\_comm\\_reply\\_t::type](#)

Definition at line 88 of file [fipa\\_comm.h](#).

The documentation for this struct was generated from the following file:

- [/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/fipa\\_comm.h](#)

## 12.50 fipa\_comm\_t Struct Reference

```
#include <fipa_comm.h>
```

### Data Fields

- [int error](#)
- [char \\* agent\\_name](#)
- [char \\* agent\\_address](#)
- [void \\* agent](#)
- [char debug](#)
- [fipa\\_comm\\_performative\\_p performative](#) [MC\_NUM\_PERFORMATIVES]
- [fipa\\_comm\\_reply\\_p default\\_reply](#)
- [fipa\\_comm\\_protocol\\_p protocol](#) [FIPA\_PROTOCOL\_END]
- [fipa\\_comm\\_callback pCallbacks](#) [FIPA\_PROTOCOL\_END]

### 12.50.1 Detailed Description

Definition at line 179 of file fipa\_comm.h.

### 12.50.2 Field Documentation

#### 12.50.2.1 void\* fipa\_comm\_t::agent

Definition at line 184 of file fipa\_comm.h.

#### 12.50.2.2 char\* fipa\_comm\_t::agent\_address

Definition at line 183 of file fipa\_comm.h.

#### 12.50.2.3 char\* fipa\_comm\_t::agent\_name

Definition at line 182 of file fipa\_comm.h.

#### 12.50.2.4 char fipa\_comm\_t::debug

Definition at line 185 of file fipa\_comm.h.

#### 12.50.2.5 fipa\_comm\_reply\_p fipa\_comm\_t::default\_reply

Definition at line 187 of file fipa\_comm.h.

#### 12.50.2.6 int fipa\_comm\_t::error

Definition at line 181 of file fipa\_comm.h.

**12.50.2.7 fipa\_comm\_callback fipa\_comm\_t::pCallbacks[FIPA\_PROTOCOL\_END]**

Definition at line 189 of file fipa\_comm.h.

**12.50.2.8 fipa\_comm\_performative\_p fipa\_comm\_t::performative[MC\_NUM\_PERFORMATIVES]**

Definition at line 186 of file fipa\_comm.h.

**12.50.2.9 fipa\_comm\_protocol\_p fipa\_comm\_t::protocol[FIPA\_PROTOCOL\_END]**

Definition at line 188 of file fipa\_comm.h.

The documentation for this struct was generated from the following file:

- /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/[fipa\\_comm.h](#)

## 12.51 fipa\_DateTime\_s Struct Reference

```
#include <fipa_acl.h>
```

### Data Fields

- [int year](#)
- [int month](#)
- [int day](#)
- [int hour](#)
- [int minute](#)
- [int second](#)
- [int millisecond](#)
- [char sign](#)
- [int is\\_utc](#)

### 12.51.1 Detailed Description

Definition at line 234 of file fipa\_acl.h.

### 12.51.2 Field Documentation

#### 12.51.2.1 int fipa\_DateTime\_s::day

Definition at line 238 of file fipa\_acl.h.

Referenced by fipa\_DateTime\_Compose().

#### 12.51.2.2 int fipa\_DateTime\_s::hour

Definition at line 239 of file fipa\_acl.h.

Referenced by fipa\_DateTime\_Compose().

#### 12.51.2.3 int fipa\_DateTime\_s::is\_utc

Definition at line 244 of file fipa\_acl.h.

#### 12.51.2.4 int fipa\_DateTime\_s::millisecond

Definition at line 242 of file fipa\_acl.h.

Referenced by fipa\_DateTime\_Compose().

#### 12.51.2.5 int fipa\_DateTime\_s::minute

Definition at line 240 of file fipa\_acl.h.

Referenced by fipa\_DateTime\_Compose().



**12.51.2.6 int fipa\_DateTime\_s::month**

Definition at line 237 of file fipa\_acl.h.

Referenced by fipa\_DateTime\_Compose().

**12.51.2.7 int fipa\_DateTime\_s::second**

Definition at line 241 of file fipa\_acl.h.

Referenced by fipa\_DateTime\_Compose().

**12.51.2.8 char fipa\_DateTime\_s::sign**

Definition at line 243 of file fipa\_acl.h.

Referenced by fipa\_DateTime\_Compose(), and fipa\_datetime\_Parse().

**12.51.2.9 int fipa\_DateTime\_s::year**

Definition at line 236 of file fipa\_acl.h.

Referenced by fipa\_DateTime\_Compose().

The documentation for this struct was generated from the following file:

- /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/fipa\_acl.h

## 12.52 fipa\_expression\_s Struct Reference

```
#include <fipa_acl.h>
```

### Data Structures

- union [content\\_u](#)

### Data Fields

- enum [fipa\\_expression\\_type\\_e](#) type
- union [fipa\\_expression\\_s::content\\_u](#) content

#### 12.52.1 Detailed Description

Definition at line 193 of file [fipa\\_acl.h](#).

#### 12.52.2 Field Documentation

##### 12.52.2.1 union [fipa\\_expression\\_s::content\\_u](#) [fipa\\_expression\\_s::content](#)

Referenced by [fipa\\_expression\\_Compose\(\)](#), [fipa\\_expression\\_Copy\(\)](#), [fipa\\_expression\\_Destroy\(\)](#), and [MC\\_AclSetConversationID\(\)](#).

##### 12.52.2.2 enum [fipa\\_expression\\_type\\_e](#) [fipa\\_expression\\_s::type](#)

Definition at line 195 of file [fipa\\_acl.h](#).

Referenced by [fipa\\_expression\\_Compose\(\)](#), [fipa\\_expression\\_Copy\(\)](#), [fipa\\_expression\\_Destroy\(\)](#), [fipa\\_expression\\_Parse\(\)](#), and [MC\\_AclSetConversationID\(\)](#).

The documentation for this struct was generated from the following file:

- [/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/fipa\\_acl.h](#)

## 12.53 fipa\_list\_t Struct Reference

```
#include <fipa_comm.h>
```

### Data Fields

- void \* [data](#)
- struct [fipa\\_list\\_t](#) \* [next](#)

### 12.53.1 Detailed Description

Definition at line 136 of file [fipa\\_comm.h](#).

### 12.53.2 Field Documentation

#### 12.53.2.1 void\* fipa\_list\_t::data

Definition at line 137 of file [fipa\\_comm.h](#).

#### 12.53.2.2 struct fipa\_list\_t\* fipa\_list\_t::next [read]

Definition at line 138 of file [fipa\\_comm.h](#).

The documentation for this struct was generated from the following file:

- [/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/fipa\\_comm.h](#)

## 12.54 fipa\_message\_string\_s Struct Reference

```
#include <fipa_acl.h>
```

### Data Fields

- char \* [message](#)
- char \* [parse](#)

### 12.54.1 Detailed Description

Definition at line 143 of file `fipa_acl.h`.

### 12.54.2 Field Documentation

#### 12.54.2.1 char\* fipa\_message\_string\_s::message

Definition at line 145 of file `fipa_acl.h`.

Referenced by `acc_connection_Thread()`, `fipa_envelope_HandleDate()`, `fipa_envelope_HandleReceived()`, `fipa_message_string_Copy()`, and `fipa_message_string_Destroy()`.

#### 12.54.2.2 char\* fipa\_message\_string\_s::parse

Definition at line 146 of file `fipa_acl.h`.

Referenced by `acc_connection_Thread()`, `fipa_agent_identifier_Parse()`, `fipa_CheckNextToken()`, `fipa_datetime_Parse()`, `fipa_envelope_HandleDate()`, `fipa_envelope_HandleReceived()`, `fipa_GetAtom()`, `fipa_GetNextWord()`, `fipa_GetWholeToken()`, `fipa_message_string_Copy()`, `fipa_string_Parse()`, and `fipa_word_Parse()`.

The documentation for this struct was generated from the following file:

- [/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/fipa\\_acl.h](#)

## 12.55 fipa\_number\_s Struct Reference

```
#include <fipa_acl.h>
```

### Data Fields

- char \* [str](#)

### 12.55.1 Detailed Description

Definition at line 264 of file fipa\_acl.h.

### 12.55.2 Field Documentation

#### 12.55.2.1 char\* fipa\_number\_s::str

Definition at line 266 of file fipa\_acl.h.

Referenced by fipa\_number\_Compose(), fipa\_number\_Copy(), and fipa\_number\_Destroy().

The documentation for this struct was generated from the following file:

- [/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/fipa\\_acl.h](#)

## 12.56 fipa\_string\_s Struct Reference

```
#include <fipa_acl.h>
```

### Data Fields

- char \* [content](#)

### 12.56.1 Detailed Description

Definition at line 223 of file fipa\_acl.h.

### 12.56.2 Field Documentation

#### 12.56.2.1 char\* fipa\_string\_s::content

Definition at line 225 of file fipa\_acl.h.

Referenced by fipa\_string\_Compose(), fipa\_string\_Copy(), fipa\_string\_Destroy(), fipa\_string\_Parse(), MC\_AclSetContent(), and MC\_AclSetConversationID().

The documentation for this struct was generated from the following file:

- [/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/fipa\\_acl.h](#)

## 12.57 fipa\_url\_s Struct Reference

```
#include <fipa_acl.h>
```

### Data Fields

- char \* [str](#)

### 12.57.1 Detailed Description

Definition at line 253 of file fipa\_acl.h.

### 12.57.2 Field Documentation

#### 12.57.2.1 char\* fipa\_url\_s::str

Definition at line 255 of file fipa\_acl.h.

Referenced by `acc_connection_Thread()`, `fipa_envelope_Compose__from()`, `fipa_envelope_Compose_-_intended_receiver()`, `fipa_envelope_Compose__to()`, `fipa_envelope_HandleReceived()`, `fipa_envelope_ParseAddresses()`, `fipa_url_Compose()`, `fipa_url_Copy()`, `fipa_url_Destroy()`, `MC_AclAddReceiver()`, `MC_AclAddReplyTo()`, `MC_AclSend()`, and `MC_AclSetSender()`.

The documentation for this struct was generated from the following file:

- `/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/fipa_acl.h`

## 12.58 fipa\_url\_sequence\_s Struct Reference

```
#include <fipa_acl.h>
```

### Data Fields

- [int num](#)
- [struct fipa\\_url\\_s \\*\\* urls](#)

### 12.58.1 Detailed Description

Definition at line 154 of file [fipa\\_acl.h](#).

### 12.58.2 Field Documentation

#### 12.58.2.1 int fipa\_url\_sequence\_s::num

Definition at line 156 of file [fipa\\_acl.h](#).

Referenced by [fipa\\_agent\\_identifier\\_Compose\(\)](#), [fipa\\_envelope\\_Compose\\_\\_from\(\)](#), [fipa\\_envelope\\_Compose\\_\\_intended\\_receiver\(\)](#), [fipa\\_envelope\\_Compose\\_\\_to\(\)](#), [fipa\\_envelope\\_ParseAddresses\(\)](#), [fipa\\_url\\_sequence\\_Compose\(\)](#), [fipa\\_url\\_sequence\\_Copy\(\)](#), [fipa\\_url\\_sequence\\_Destroy\(\)](#), [MC\\_AclAddReceiver\(\)](#), [MC\\_AclAddReplyTo\(\)](#), [MC\\_AclSend\(\)](#), and [MC\\_AclSetSender\(\)](#).

#### 12.58.2.2 struct fipa\_url\_s\*\* fipa\_url\_sequence\_s::urls [read]

Definition at line 157 of file [fipa\\_acl.h](#).

Referenced by [acc\\_connection\\_Thread\(\)](#), [fipa\\_envelope\\_Compose\\_\\_from\(\)](#), [fipa\\_envelope\\_Compose\\_\\_intended\\_receiver\(\)](#), [fipa\\_envelope\\_Compose\\_\\_to\(\)](#), [fipa\\_envelope\\_ParseAddresses\(\)](#), [fipa\\_url\\_sequence\\_Compose\(\)](#), [fipa\\_url\\_sequence\\_Copy\(\)](#), [fipa\\_url\\_sequence\\_Destroy\(\)](#), [MC\\_AclAddReceiver\(\)](#), [MC\\_AclAddReplyTo\(\)](#), [MC\\_AclSend\(\)](#), and [MC\\_AclSetSender\(\)](#).

The documentation for this struct was generated from the following file:

- [/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/fipa\\_acl.h](#)



## 12.59 fipa\_word\_s Struct Reference

```
#include <fipa_acl.h>
```

### Data Fields

- char \* [content](#)

### 12.59.1 Detailed Description

Definition at line 212 of file fipa\_acl.h.

### 12.59.2 Field Documentation

#### 12.59.2.1 char\* fipa\_word\_s::content

Definition at line 214 of file fipa\_acl.h.

Referenced by fipa\_agent\_identifier\_Parse(), fipa\_agent\_identifier\_set\_Parse(), fipa\_message\_parameter\_Parse(), fipa\_message\_type\_Parse(), fipa\_protocol\_type\_Parse(), fipa\_url\_Parse(), fipa\_url\_sequence\_Parse(), fipa\_word\_Compose(), fipa\_word\_Copy(), and fipa\_word\_Destroy().

The documentation for this struct was generated from the following file:

- [/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/fipa\\_acl.h](#)

## 12.60 foo\_c Class Reference

### Public Member Functions

- [foo\\_c](#) (float *f*, int *b*)
- [~foo\\_c](#) ()
- [int get\\_bar](#) ()
- float [get\\_foo](#) ()
- void [set\\_bar](#) (int *b*)
- void [set\\_foo](#) (float *f*)
- void [set\\_foobar](#) (float *f*, int *b*=0)
- [int get\\_barfoo](#) ()

### Protected Member Functions

- [int get\\_global](#) ()

### Static Protected Attributes

- static [int global](#)

### Private Attributes

- float [foo](#)
- [int bar](#)
- [int barfoo](#)

#### 12.60.1 Detailed Description

Definition at line 1 of file class.cxx.

#### 12.60.2 Constructor & Destructor Documentation

##### 12.60.2.1 foo\_c::foo\_c (float *f*, int *b*)

##### 12.60.2.2 foo\_c::~~foo\_c ()

Definition at line 81 of file class.cxx.

#### 12.60.3 Member Function Documentation

##### 12.60.3.1 int foo\_c::get\_bar () [inline]

Definition at line 13 of file class.cxx.

References [bar](#).

**12.60.3.2 int foo\_c::get\_barfoo () [inline]**

Definition at line 67 of file class.cxx.

References barfoo.

**12.60.3.3 float foo\_c::get\_foo () [inline]**

Definition at line 20 of file class.cxx.

References foo.

**12.60.3.4 int foo\_c::get\_global () [inline, protected]**

Definition at line 54 of file class.cxx.

References global.

**12.60.3.5 void foo\_c::set\_bar (int *b*) [inline]**

Definition at line 27 of file class.cxx.

References bar.

**12.60.3.6 void foo\_c::set\_foo (float *f*) [inline]**

Definition at line 34 of file class.cxx.

References foo.

**12.60.3.7 void foo\_c::set\_foobar (float *f*, int *b* = 0) [inline]**

Definition at line 41 of file class.cxx.

References bar, and foo.

**12.60.4 Field Documentation****12.60.4.1 int foo\_c::bar [private]**

Definition at line 4 of file class.cxx.

Referenced by get\_bar(), set\_bar(), and set\_foobar().

**12.60.4.2 int foo\_c::barfoo [private]**

Definition at line 61 of file class.cxx.

Referenced by get\_barfoo().

**12.60.4.3 float foo\_c::foo [private]**

Definition at line 3 of file class.cxx.

Referenced by `get_foo()`, `set_foo()`, and `set_foobar()`.

**12.60.4.4 int foo\_c::global [static, protected]**

Definition at line 50 of file class.cxx.

Referenced by `get_global()`.

The documentation for this class was generated from the following file:

- `/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mxml-2.2.2/test/class.cxx`

## 12.61 foo\_s Struct Reference

### Public Member Functions

- [foo\\_s](#) (float [f](#), int [b](#))
- [~foo\\_s](#) ()
- int [get\\_bar](#) ()
- float [get\\_foo](#) ()
- void [set\\_bar](#) (int [b](#))
- void [set\\_foo](#) (float [f](#))

### Data Fields

- float [foo](#)
- int [bar](#)

#### 12.61.1 Detailed Description

Definition at line 1 of file struct.cxx.

#### 12.61.2 Constructor & Destructor Documentation

##### 12.61.2.1 foo\_s::foo\_s (float *f*, int *b*)

Definition at line 39 of file struct.cxx.

References [bar](#), and [foo](#).

##### 12.61.2.2 foo\_s::~~foo\_s ()

Definition at line 46 of file struct.cxx.

#### 12.61.3 Member Function Documentation

##### 12.61.3.1 int foo\_s::get\_bar () [[inline](#)]

Definition at line 11 of file struct.cxx.

References [bar](#).

##### 12.61.3.2 float foo\_s::get\_foo () [[inline](#)]

Definition at line 18 of file struct.cxx.

References [foo](#).

**12.61.3.3 void foo\_s::set\_bar (int *b*) [inline]**

Definition at line 25 of file struct.cxx.

References bar.

**12.61.3.4 void foo\_s::set\_foo (float *f*) [inline]**

Definition at line 32 of file struct.cxx.

References foo.

**12.61.4 Field Documentation****12.61.4.1 int foo\_s::bar**

Definition at line 4 of file struct.cxx.

Referenced by foo\_s(), get\_bar(), and set\_bar().

**12.61.4.2 float foo\_s::foo**

Definition at line 3 of file struct.cxx.

Referenced by foo\_s(), get\_foo(), and set\_foo().

The documentation for this struct was generated from the following file:

- /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mxml-2.2.2/test/[struct.cxx](#)

## 12.62 havege\_state Struct Reference

HAVEGE state structure.

```
#include <havege.h>
```

### Data Fields

- [int PT1](#)
- [int PT2](#)
- [int offset](#) [2]
- [int pool](#) [COLLECT\_SIZE]
- [int WALK](#) [8192]

### 12.62.1 Detailed Description

HAVEGE state structure.

Definition at line 12 of file havege.h.

### 12.62.2 Field Documentation

#### 12.62.2.1 int havege\_state::offset[2]

Definition at line 14 of file havege.h.

Referenced by `havege_fill()`, and `havege_rand()`.

#### 12.62.2.2 int havege\_state::pool[COLLECT\_SIZE]

Definition at line 15 of file havege.h.

Referenced by `havege_rand()`.

#### 12.62.2.3 int havege\_state::PT1

Definition at line 14 of file havege.h.

Referenced by `havege_fill()`.

#### 12.62.2.4 int havege\_state::PT2

Definition at line 14 of file havege.h.

Referenced by `havege_fill()`.

#### 12.62.2.5 int havege\_state::WALK[8192]

Definition at line 16 of file havege.h.

Referenced by `havege_fill()`.

The documentation for this struct was generated from the following file:

- </home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/havege.h>



## 12.63 host\_id\_s Struct Reference

```
#include <host_id.h>
```

### Data Fields

- char \* [hostname](#)
- [int](#) port

### 12.63.1 Detailed Description

Definition at line 38 of file host\_id.h.

### 12.63.2 Field Documentation

#### 12.63.2.1 char\* host\_id\_s::hostname

Definition at line 40 of file host\_id.h.

#### 12.63.2.2 int host\_id\_s::port

Definition at line 41 of file host\_id.h.

The documentation for this struct was generated from the following file:

- /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/[host\\_id.h](#)

## 12.64 hr\_time Struct Reference

timer structure

```
#include <timing.h>
```

### Data Fields

- unsigned char [opaque](#) [32]

### 12.64.1 Detailed Description

timer structure

Definition at line 10 of file `timing.h`.

### 12.64.2 Field Documentation

#### 12.64.2.1 unsigned char hr\_time::opaque[32]

Definition at line 12 of file `timing.h`.

The documentation for this struct was generated from the following file:

- `/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/timing.h`

## 12.65 interpreter\_variable\_data\_s Struct Reference

```
#include <interpreter_variable_data.h>
```

### Data Fields

- char \* [name](#)
- int [size](#)
- ChType\_t [data\\_type](#)
- int [array\\_dim](#)
- int \* [array\\_extent](#)
- void \* [data](#)

### 12.65.1 Detailed Description

Definition at line 41 of file `interpreter_variable_data.h`.

### 12.65.2 Field Documentation

#### 12.65.2.1 int interpreter\_variable\_data\_s::array\_dim

Definition at line 45 of file `interpreter_variable_data.h`.

Referenced by `agent_AddPersistentVariable()`, `agent_xml_compose__data()`, `agent_xml_compose__row()`, `agent_xml_parse__data()`, `agent_xml_parse__row()`, `interpreter_variable_data_Copy()`, `interpreter_variable_data_Initialize()`, `interpreter_variable_data_InitializeFromAgent()`, `interpreter_variable_data_New()`, `MC_AgentVariableRetrieveInfo()`, and `MC_GetAgentReturnData()`.

#### 12.65.2.2 int\* interpreter\_variable\_data\_s::array\_extent

Definition at line 46 of file `interpreter_variable_data.h`.

Referenced by `agent_AddPersistentVariable()`, `agent_xml_compose__row()`, `agent_xml_parse__row()`, `interpreter_variable_data_Copy()`, `interpreter_variable_data_Destroy()`, `interpreter_variable_data_Initialize()`, `interpreter_variable_data_InitializeFromAgent()`, `interpreter_variable_data_New()`, `MC_AgentVariableRetrieveInfo()`, and `MC_GetAgentReturnData()`.

#### 12.65.2.3 void\* interpreter\_variable\_data\_s::data

Definition at line 47 of file `interpreter_variable_data.h`.

Referenced by `agent_AddPersistentVariable()`, `agent_xml_compose__data()`, `agent_xml_compose__row()`, `agent_xml_parse__data()`, `agent_xml_parse__row()`, `interpreter_variable_data_Copy()`, `interpreter_variable_data_Destroy()`, `interpreter_variable_data_Initialize()`, `interpreter_variable_data_InitializeFromAgent()`, `interpreter_variable_data_New()`, `MC_AgentVariableRetrieveInfo()`, `MC_AgentVariableRetrieveInfo()`, and `MC_SaveData_chdl()`.

#### 12.65.2.4 ChType\_t interpreter\_variable\_data\_s::data\_type

Definition at line 44 of file interpreter\_variable\_data.h.

Referenced by agent\_AddPersistentVariable(), agent\_xml\_compose\_\_data(), agent\_xml\_compose\_\_row(), agent\_xml\_parse\_\_data(), agent\_xml\_parse\_\_row(), interpreter\_variable\_data\_Copy(), interpreter\_variable\_data\_Initialize(), interpreter\_variable\_data\_InitializeFromAgent(), interpreter\_variable\_data\_New(), MC\_GetAgentReturnData(), and MC\_SaveData\_chdl().

#### 12.65.2.5 char\* interpreter\_variable\_data\_s::name

Definition at line 42 of file interpreter\_variable\_data.h.

Referenced by agent\_AddPersistentVariable(), agent\_xml\_compose\_\_data(), agent\_xml\_parse\_\_data(), interpreter\_variable\_data\_Copy(), interpreter\_variable\_data\_Destroy(), interpreter\_variable\_data\_Initialize(), interpreter\_variable\_data\_InitializeFromAgent(), interpreter\_variable\_data\_New(), and MC\_SaveData\_chdl().

#### 12.65.2.6 int interpreter\_variable\_data\_s::size

Definition at line 43 of file interpreter\_variable\_data.h.

Referenced by agent\_AddPersistentVariable(), interpreter\_variable\_data\_Copy(), interpreter\_variable\_data\_Initialize(), interpreter\_variable\_data\_InitializeFromAgent(), interpreter\_variable\_data\_New(), and MC\_SaveData\_chdl().

The documentation for this struct was generated from the following file:

- /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/[interpreter\\_variable\\_data.h](#)

## 12.66 LibMC::InvalidAgencyException Class Reference

Exception class for use with null agency pointers.

### Public Member Functions

- [InvalidAgencyException](#) ()  
*Null agency pointer exception constructor.*
- [InvalidAgencyException](#) (String exc)  
*Null agency pointer exception constructor.*

### Private Attributes

- const String [msg](#) = "Private agency pointer not valid."

#### 12.66.1 Detailed Description

Exception class for use with null agency pointers. This class provides a way to inform the program that an agency was created or accessed that had an invalid internal pointer.

Definition at line 1074 of file MCAgency.cs.

#### 12.66.2 Constructor & Destructor Documentation

##### 12.66.2.1 LibMC::InvalidAgencyException::InvalidAgencyException () [inline]

Null agency pointer exception constructor. Constructor for the exception class. This exception simply defines a recognizable exception class and sets the exception message appropriately.

Definition at line 1085 of file MCAgency.cs.

##### 12.66.2.2 LibMC::InvalidAgencyException::InvalidAgencyException (String exc) [inline]

Null agency pointer exception constructor. Constructor for the exception class. Allows the use of a custom error message.

Definition at line 1096 of file MCAgency.cs.

#### 12.66.3 Field Documentation

##### 12.66.3.1 const String LibMC::InvalidAgencyException::msg = "Private agency pointer not valid." [private]

Definition at line 1076 of file MCAgency.cs.

The documentation for this class was generated from the following file:

- </home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/win32/LibMC.net/LibMC/MCAgency.cs>

## 12.67 LibMC::InvalidAgentException Class Reference

Exception class for use with null agent pointers.

### Public Member Functions

- [InvalidAgentException \(\)](#)  
*Null agent pointer exception constructor.*
- [InvalidAgentException \(String exc\)](#)  
*Null agent pointer exception constructor.*

### Private Attributes

- const String [msg](#) = "Private agent pointer not valid."

#### 12.67.1 Detailed Description

Exception class for use with null agent pointers. This class provides a way to inform the program that an agent was created or accessed that had an invalid internal pointer.

Definition at line 568 of file MCAgent.cs.

#### 12.67.2 Constructor & Destructor Documentation

##### 12.67.2.1 LibMC::InvalidAgentException::InvalidAgentException () **[inline]**

Null agent pointer exception constructor. Constructor for the exception class. This exception simply defines a recognizable exception class and sets the exception message appropriately.

Definition at line 579 of file MCAgent.cs.

##### 12.67.2.2 LibMC::InvalidAgentException::InvalidAgentException (String exc) **[inline]**

Null agent pointer exception constructor. Constructor for the exception class. Allows the use of a custom error message.

Definition at line 590 of file MCAgent.cs.

#### 12.67.3 Field Documentation

##### 12.67.3.1 const String LibMC::InvalidAgentException::msg = "Private agent pointer not valid." **[private]**

Definition at line 570 of file MCAgent.cs.

The documentation for this class was generated from the following file:

- </home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/win32/LibMC.net/LibMC/MCAgent.cs>

## 12.68 list\_s Struct Reference

```
#include <list.h>
```

### Data Fields

- [listNode\\_p listhead](#)
- [int size](#)

### 12.68.1 Detailed Description

Definition at line 41 of file list.h.

### 12.68.2 Field Documentation

#### 12.68.2.1 listNode\_p list\_s::listhead

Definition at line 43 of file list.h.

Referenced by `barrier_queue_Add()`, `barrier_queue_Get()`, `ListAdd()`, `ListDelete()`, `ListGetHead()`, `ListInitialize()`, `ListPop()`, `ListSearch()`, `ListTerminate()`, `syncListAddNode()`, and `syncListFind()`.

#### 12.68.2.2 int list\_s::size

Definition at line 44 of file list.h.

Referenced by `barrier_queue_Delete()`, `list_pGetSize()`, `ListAdd()`, `ListDelete()`, `ListInitialize()`, `ListPop()`, `ListSearch()`, `ListTerminate()`, `syncListDelete()`, and `syncListRemove()`.

The documentation for this struct was generated from the following file:

- `/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mc_list/list.h`

## 12.69 listNode\_s Struct Reference

```
#include <list.h>
```

### Data Fields

- DATA [node\\_data](#)
- struct [listNode\\_s](#) \* [next](#)

### 12.69.1 Detailed Description

Definition at line 32 of file list.h.

### 12.69.2 Field Documentation

#### 12.69.2.1 struct listNode\_s\* listNode\_s::next [read]

Definition at line 35 of file list.h.

Referenced by `AP_QUEUE_STD_DEFN_TEMPLATE()`, `barrier_queue_Add()`, `barrier_queue_Get()`, `df_SearchForService()`, `ListAdd()`, `ListDelete()`, `ListPop()`, `ListSearch()`, `request_handler_DEREGISTER()`, `syncListAddNode()`, and `syncListFind()`.

#### 12.69.2.2 DATA listNode\_s::node\_data

Definition at line 34 of file list.h.

Referenced by `AP_QUEUE_STD_DEFN_TEMPLATE()`, `barrier_queue_Add()`, `barrier_queue_Get()`, `df_SearchForService()`, `ListAdd()`, `ListDelete()`, `ListGetHead()`, `ListPop()`, `ListSearch()`, `request_handler_DEREGISTER()`, `syncListAddNode()`, and `syncListFind()`.

The documentation for this struct was generated from the following file:

- `/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mc_list/list.h`



## 12.70 mc\_platform\_s Struct Reference

```
#include <mc_platform.h>
```

### Data Fields

- [int err](#)
- [char \\* hostname](#)
- [unsigned short port](#)
- [int initInterps](#)
- [message\\_queue\\_p asm\\_message\\_queue](#)
- [message\\_queue\\_p message\\_queue](#)
- [agent\\_queue\\_p agent\\_queue](#)
- [connection\\_queue\\_p connection\\_queue](#)
- [df\\_p df](#)
- [ams\\_p ams](#)
- [acc\\_p acc](#)
- [cmd\\_prompt\\_p cmd\\_prompt](#)
- [syncList\\_p syncList](#)
- [barrier\\_queue\\_p barrier\\_queue](#)
- [listen\\_thread\\_arg\\_p listen\\_thread\\_arg](#)
- [listen\\_thread\\_arg\\_p client\\_thread\\_arg](#)
- [int default\\_agentstatus](#)
- [int stack\\_size \[MC\\_THREAD\\_ALL\]](#)
- [ChOptions\\_t \\* interp\\_options](#)
- [COND\\_T \\* MC\\_signal\\_cond](#)
- [COND\\_T \\* MC\\_sync\\_cond](#)
- [MUTEX\\_T \\* MC\\_signal\\_lock](#)
- [MUTEX\\_T \\* MC\\_sync\\_lock](#)
- [enum MC\\_Signal\\_e MC\\_signal](#)
- [enum MC\\_SteerCommand\\_e MC\\_steer\\_command](#)
- [MUTEX\\_T \\* MC\\_steer\\_lock](#)
- [COND\\_T \\* MC\\_steer\\_cond](#)
- [int giant](#)
- [MUTEX\\_T \\* giant\\_lock](#)
- [COND\\_T \\* giant\\_cond](#)
- [int quit](#)
- [MUTEX\\_T \\* quit\\_lock](#)
- [COND\\_T \\* quit\\_cond](#)
- [int sockfd](#)
- [char private\\_key \[1210\]](#)
- [MCAgency\\_t agency](#)
- [interpreter\\_queue\\_p interpreter\\_queue](#)

### 12.70.1 Detailed Description

Definition at line 49 of file mc\_platform.h.

## 12.70.2 Field Documentation

### 12.70.2.1 `acc_p mc_platform_s::acc`

Definition at line 73 of file `mc_platform.h`.

Referenced by `acc_connection_Thread()`, `acc_MessageHandlerThread()`, `acc_Start()`, `acc_Thread()`, `listen_Thread()`, `MC_End()`, `mc_platform_Destroy()`, and `message_Send()`.

### 12.70.2.2 `MCAgency_t mc_platform_s::agency`

Definition at line 115 of file `mc_platform.h`.

Referenced by `listen_Thread()`, `MC_Initialize()`, and `message_send_Thread()`.

### 12.70.2.3 `agent_queue_p mc_platform_s::agent_queue`

Definition at line 68 of file `mc_platform.h`.

Referenced by `acc_connection_Thread()`, `acc_MessageHandlerThread()`, `handler_FLUSH_AGENTS()`, `handler_PRINTLIST_AGENTS()`, `MC_AddAgent()`, `MC_AddStationaryAgent()`, `MC_FindAgentByID()`, `MC_FindAgentByName()`, `MC_GetAllAgents()`, `mc_platform_Destroy()`, `MC_RetrieveAgent()`, `MC_SendAgentMigrationMessageFile()`, `MC_WaitAgent()`, and `MC_WaitRetrieveAgent()`.

### 12.70.2.4 `ams_p mc_platform_s::ams`

Definition at line 72 of file `mc_platform.h`.

Referenced by `acc_MessageHandlerThread()`, `agent_RunChScriptThread()`, `ams_Start()`, `ams_Thread()`, `MC_AddAgent()`, `MC_End()`, `mc_platform_Destroy()`, `MC_SendAgentMigrationMessageFile()`, and `MC_SetAgentStatus()`.

### 12.70.2.5 `message_queue_p mc_platform_s::asm_message_queue`

Definition at line 63 of file `mc_platform.h`.

### 12.70.2.6 `barrier_queue_p mc_platform_s::barrier_queue`

Definition at line 80 of file `mc_platform.h`.

Referenced by `MC_Barrier()`, `MC_BarrierDelete()`, `MC_BarrierInit()`, and `mc_platform_Destroy()`.

### 12.70.2.7 `listen_thread_arg_p mc_platform_s::client_thread_arg`

Definition at line 83 of file `mc_platform.h`.

### 12.70.2.8 `cmd_prompt_p mc_platform_s::cmd_prompt`

Definition at line 74 of file `mc_platform.h`.

Referenced by `cmd_prompt_Start()`, `MC_End()`, and `mc_platform_Destroy()`.

**12.70.2.9 connection\_queue\_p mc\_platform\_s::connection\_queue**

Definition at line 69 of file mc\_platform.h.

Referenced by acc\_Thread(), handler\_PRINT\_CONNECTLIST(), listen\_Thread(), MC\_End(), and mc\_platform\_Destroy().

**12.70.2.10 int mc\_platform\_s::default\_agentstatus**

Definition at line 85 of file mc\_platform.h.

Referenced by agent\_Initialize(), and MC\_SetDefaultAgentStatus().

**12.70.2.11 df\_p mc\_platform\_s::df**

Definition at line 71 of file mc\_platform.h.

Referenced by df\_ProcessRequest(), df\_Start(), df\_Thread(), MC\_DeregisterService(), MC\_End(), mc\_platform\_Destroy(), MC\_RegisterService(), MC\_SearchForService(), request\_handler\_DEREGISTER(), request\_handler\_REGISTER(), and request\_handler\_SEARCH().

**12.70.2.12 int mc\_platform\_s::err**

Definition at line 53 of file mc\_platform.h.

Referenced by agent\_Initialize(), message\_InitializeFromAgent(), and message\_InitializeFromString().

**12.70.2.13 int mc\_platform\_s::giant**

Definition at line 105 of file mc\_platform.h.

Referenced by acc\_MessageHandlerThread(), acc\_Thread(), MC\_GetAllAgents(), MC\_HaltAgency(), MC\_ResetSignal(), MC\_ResumeAgency(), and MC\_WaitSignal().

**12.70.2.14 COND\_T\* mc\_platform\_s::giant\_cond**

Definition at line 107 of file mc\_platform.h.

Referenced by acc\_MessageHandlerThread(), acc\_Thread(), mc\_platform\_Destroy(), and MC\_ResetSignal().

**12.70.2.15 MUTEX\_T\* mc\_platform\_s::giant\_lock**

Definition at line 106 of file mc\_platform.h.

Referenced by acc\_MessageHandlerThread(), acc\_Thread(), MC\_GetAllAgents(), MC\_HaltAgency(), mc\_platform\_Destroy(), MC\_ResetSignal(), MC\_ResumeAgency(), and MC\_WaitSignal().

**12.70.2.16 char\* mc\_platform\_s::hostname**

Definition at line 58 of file mc\_platform.h.

Referenced by `agent_Initialize()`, `agent_NewBinary()`, `fipa_envelope_Compose__from()`, `message_InitializeFromAgent()`, `message_InitializeFromString()`, `message_queue_SendOutgoing()`, and `udplisten_Thread()`.

#### **12.70.2.17 int mc\_platform\_s::initInterps**

Definition at line 60 of file `mc_platform.h`.

#### **12.70.2.18 ChOptions\_t\* mc\_platform\_s::interp\_options**

Definition at line 89 of file `mc_platform.h`.

Referenced by `agent_RunChScriptThread()`, and `mc_platform_Destroy()`.

#### **12.70.2.19 interpreter\_queue\_p mc\_platform\_s::interpreter\_queue**

Definition at line 118 of file `mc_platform.h`.

Referenced by `agent_Destroy()`, and `agent_RunChScriptThread()`.

#### **12.70.2.20 listen\_thread\_arg\_p mc\_platform\_s::listen\_thread\_arg**

Definition at line 82 of file `mc_platform.h`.

#### **12.70.2.21 enum MC\_Signal\_e mc\_platform\_s::MC\_signal**

Definition at line 96 of file `mc_platform.h`.

Referenced by `acc_MessageHandlerThread()`, `acc_Thread()`, `agent_RunChScriptThread()`, `MC_ResetSignal()`, and `MC_WaitSignal()`.

#### **12.70.2.22 COND\_T\* mc\_platform\_s::MC\_signal\_cond**

Definition at line 92 of file `mc_platform.h`.

Referenced by `acc_MessageHandlerThread()`, `acc_Thread()`, `agent_RunChScriptThread()`, `mc_platform_Destroy()`, and `MC_WaitSignal()`.

#### **12.70.2.23 MUTEX\_T\* mc\_platform\_s::MC\_signal\_lock**

Definition at line 94 of file `mc_platform.h`.

Referenced by `acc_MessageHandlerThread()`, `acc_Thread()`, `agent_RunChScriptThread()`, `mc_platform_Destroy()`, and `MC_WaitSignal()`.

#### **12.70.2.24 enum MC\_SteerCommand\_e mc\_platform\_s::MC\_steer\_command**

Definition at line 99 of file `mc_platform.h`.

Referenced by `MC_SendSteerCommand()`, `MC_Steer()`, and `MC_SteerControl()`.

**12.70.2.25 COND\_T\* mc\_platform\_s::MC\_steer\_cond**

Definition at line 101 of file mc\_platform.h.

Referenced by mc\_platform\_Destroy(), MC\_SendSteerCommand(), and MC\_SteerControl().

**12.70.2.26 MUTEX\_T\* mc\_platform\_s::MC\_steer\_lock**

Definition at line 100 of file mc\_platform.h.

Referenced by mc\_platform\_Destroy(), MC\_SendSteerCommand(), MC\_Steer(), and MC\_SteerControl().

**12.70.2.27 COND\_T\* mc\_platform\_s::MC\_sync\_cond**

Definition at line 93 of file mc\_platform.h.

Referenced by mc\_platform\_Destroy().

**12.70.2.28 MUTEX\_T\* mc\_platform\_s::MC\_sync\_lock**

Definition at line 95 of file mc\_platform.h.

Referenced by mc\_platform\_Destroy().

**12.70.2.29 message\_queue\_p mc\_platform\_s::message\_queue**

Definition at line 67 of file mc\_platform.h.

Referenced by acc\_connection\_Thread(), acc\_MessageHandlerThread(), handler\_PRINTLIST\_MESSAGE(), MC\_End(), MC\_LoadAgentFromFile(), mc\_platform\_Destroy(), and MC\_SendAgentMigrationMessage().

**12.70.2.30 unsigned short mc\_platform\_s::port**

Definition at line 59 of file mc\_platform.h.

Referenced by acc\_connection\_Thread(), agent\_Initialize(), agent\_NewBinary(), fipa\_envelope\_Compose\_\_from(), listen\_Thread(), message\_InitializeFromAgent(), message\_InitializeFromString(), message\_queue\_SendOutgoing(), and udplisten\_Thread().

**12.70.2.31 char mc\_platform\_s::private\_key[1210]**

Definition at line 113 of file mc\_platform.h.

Referenced by acc\_connection\_Thread(), listen\_Thread(), and MC\_AclSend().

**12.70.2.32 int mc\_platform\_s::quit**

Definition at line 109 of file mc\_platform.h.

Referenced by acc\_MessageHandlerThread(), acc\_Thread(), ams\_ManageAgentList(), ams\_Thread(), df\_Thread(), handler\_QUIT(), MC\_End(), and MC\_MainLoop().

**12.70.2.33 COND\_T\* mc\_platform\_s::quit\_cond**

Definition at line 111 of file mc\_platform.h.

Referenced by handler\_QUIT(), MC\_End(), and MC\_MainLoop().

**12.70.2.34 MUTEX\_T\* mc\_platform\_s::quit\_lock**

Definition at line 110 of file mc\_platform.h.

Referenced by acc\_MessageHandlerThread(), acc\_Thread(), ams\_ManageAgentList(), ams\_Thread(), df\_Thread(), handler\_QUIT(), MC\_End(), MC\_MainLoop(), and mc\_platform\_Destroy().

**12.70.2.35 int mc\_platform\_s::sockfd**

Definition at line 112 of file mc\_platform.h.

Referenced by listen\_Thread(), and mc\_platform\_Destroy().

**12.70.2.36 int mc\_platform\_s::stack\_size[MC\_THREAD\_ALL]**

Definition at line 87 of file mc\_platform.h.

Referenced by acc\_Start(), agent\_RunChScript(), ams\_Start(), cmd\_prompt\_Start(), and df\_Start().

**12.70.2.37 syncList\_p mc\_platform\_s::syncList**

Definition at line 79 of file mc\_platform.h.

Referenced by MC\_CondBroadcast(), MC\_CondReset(), MC\_CondSignal(), MC\_CondWait(), MC\_MutexLock(), MC\_MutexUnlock(), MC\_SemaphorePost(), MC\_SemaphoreWait(), MC\_SyncDelete(), and MC\_SyncInit().

The documentation for this struct was generated from the following file:

- /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/[mc\\_platform.h](#)

## 12.71 mc\_rwlock\_s Struct Reference

```
#include <mc_rwlock.h>
```

### Data Fields

- [int num\\_readers](#)
- [int write\\_flag](#)
- [int write\\_request](#)
- [MUTEX\\_T \\* lock](#)
- [COND\\_T \\* cond](#)

#### 12.71.1 Detailed Description

Definition at line 39 of file mc\_rwlock.h.

#### 12.71.2 Field Documentation

##### 12.71.2.1 COND\_T\* mc\_rwlock\_s::cond

Definition at line 44 of file mc\_rwlock.h.

Referenced by [mc\\_rwlock\\_destroy\(\)](#), [mc\\_rwlock\\_init\(\)](#), [mc\\_rwlock\\_rdlock\(\)](#), [mc\\_rwlock\\_rdunlock\(\)](#), [mc\\_rwlock\\_wrlock\(\)](#), and [mc\\_rwlock\\_wrunlock\(\)](#).

##### 12.71.2.2 MUTEX\_T\* mc\_rwlock\_s::lock

Definition at line 43 of file mc\_rwlock.h.

Referenced by [mc\\_rwlock\\_destroy\(\)](#), [mc\\_rwlock\\_init\(\)](#), [mc\\_rwlock\\_rdlock\(\)](#), [mc\\_rwlock\\_rdunlock\(\)](#), [mc\\_rwlock\\_wrlock\(\)](#), and [mc\\_rwlock\\_wrunlock\(\)](#).

##### 12.71.2.3 int mc\_rwlock\_s::num\_readers

Definition at line 40 of file mc\_rwlock.h.

Referenced by [mc\\_rwlock\\_init\(\)](#), [mc\\_rwlock\\_rdlock\(\)](#), [mc\\_rwlock\\_rdunlock\(\)](#), and [mc\\_rwlock\\_wrlock\(\)](#).

##### 12.71.2.4 int mc\_rwlock\_s::write\_flag

Definition at line 41 of file mc\_rwlock.h.

Referenced by [mc\\_rwlock\\_init\(\)](#), [mc\\_rwlock\\_rdlock\(\)](#), [mc\\_rwlock\\_wrlock\(\)](#), and [mc\\_rwlock\\_wrunlock\(\)](#).

##### 12.71.2.5 int mc\_rwlock\_s::write\_request

Definition at line 42 of file mc\_rwlock.h.

Referenced by [mc\\_rwlock\\_init\(\)](#), [mc\\_rwlock\\_rdlock\(\)](#), and [mc\\_rwlock\\_wrlock\(\)](#).

The documentation for this struct was generated from the following file:

- [/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/mc\\_rwlock.h](/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/mc_rwlock.h)



## 12.72 LibMC::MCAclMessage Class Reference

Encapsulates ACL messages in the Mobile-C library.

### Public Types

- enum [MC\\_FipaPerformative\\_e](#) {  
    [FIPA\\_ERROR](#) = -1, [FIPA\\_ZERO](#), [FIPA\\_ACCEPT\\_PROPOSAL](#), [FIPA\\_AGREE](#),  
    [FIPA\\_CANCEL](#), [FIPA\\_CALL\\_FOR\\_PROPOSAL](#), [FIPA\\_CONFIRM](#), [FIPA\\_DISCONFIRM](#),  
    [FIPA\\_FAILURE](#), [FIPA\\_INFORM](#), [FIPA\\_INFORM\\_IF](#), [FIPA\\_INFORM\\_REF](#),  
    [FIPA\\_NOT\\_UNDERSTOOD](#), [FIPA\\_PROPOGATE](#), [FIPA\\_PROPOSE](#), [FIPA\\_PROXY](#),  
    [FIPA\\_QUERY\\_IF](#), [FIPA\\_QUERY\\_REF](#), [FIPA\\_REFUSE](#), [FIPA\\_REJECT\\_PROPOSAL](#),  
    [FIPA\\_REQUEST](#), [FIPA\\_REQUEST\\_WHEN](#), [FIPA\\_REQUEST\\_WHENEVER](#), [FIPA\\_-](#)  
    [SUBSCRIBE](#) }

*Enum for describing the type of an ACL message.*

### Public Member Functions

- [MCAclMessage](#) ()  
*Default constructor.*
- void [New](#) ()  
*Creates a new, blank ACL message.*
- [MCAclMessage Reply](#) ([MCAclMessage](#) acl\_message)  
*Creates an ACL message that is a response to the argument.*
- int [SetPerformative](#) ([MC\\_FipaPerformative\\_e](#) performative)  
*Sets the performative field of the message.*
- int [SetSender](#) (String name, String address)  
*Sets the sender field of the message.*
- int [AddReceiver](#) (String name, String address)  
*Adds a receiver to the list of receivers.*
- int [AddReplyTo](#) (String name, String address)  
*Adds a "reply-to" field to the message.*
- int [SetContent](#) (String content)  
*Sets the content field of the message.*
- int [Destroy](#) ()  
*Destroys a message.*

## Properties

- internal IntPtr [AclMsg](#) [get, set]

## Private Member Functions

- internal [MCAclMessage](#) (IntPtr msg)

## Private Attributes

- IntPtr [aclmsg\\_p](#)

### 12.72.1 Detailed Description

Encapsulates ACL messages in the Mobile-C library. This class contains a pointer to an ACL message in the Mobile-C library. Functions are provided to send the message, set its various fields, and destroy the message.

Definition at line 72 of file MCAclMessage.cs.

### 12.72.2 Member Enumeration Documentation

#### 12.72.2.1 enum LibMC::MCAclMessage::MC\_FipaPerformative\_e

Enum for describing the type of an ACL message.

##### Note:

This enum is pulled directly from the Mobile-C library.

##### Enumerator:

**FIPA\_ERROR** Fipa performative enum value  
**FIPA\_ZERO** Fipa performative enum value  
**FIPA\_ACCEPT\_PROPOSAL** Fipa performative enum value  
**FIPA\_AGREE** Fipa performative enum value  
**FIPA\_CANCEL** Fipa performative enum value  
**FIPA\_CALL\_FOR\_PROPOSAL** Fipa performative enum value  
**FIPA\_CONFIRM** Fipa performative enum value  
**FIPA\_DISCONFIRM** Fipa performative enum value  
**FIPA\_FAILURE** Fipa performative enum value  
**FIPA\_INFORM** Fipa performative enum value  
**FIPA\_INFORM\_IF** Fipa performative enum value  
**FIPA\_INFORM\_REF** Fipa performative enum value  
**FIPA\_NOT\_UNDERSTOOD** Fipa performative enum value  
**FIPA\_PROPOGATE** Fipa performative enum value  
**FIPA\_PROPOSE** Fipa performative enum value

*FIPA\_PROXY* Fipa performative enum value  
*FIPA\_QUERY\_IF* Fipa performative enum value  
*FIPA\_QUERY\_REF* Fipa performative enum value  
*FIPA\_REFUSE* Fipa performative enum value  
*FIPA\_REJECT\_PROPOSAL* Fipa performative enum value  
*FIPA\_REQUEST* Fipa performative enum value  
*FIPA\_REQUEST\_WHEN* Fipa performative enum value  
*FIPA\_REQUEST\_WHENEVER* Fipa performative enum value  
*FIPA\_SUBSCRIBE* Fipa performative enum value

Definition at line 79 of file MCAclMessage.cs.

### 12.72.3 Constructor & Destructor Documentation

#### 12.72.3.1 LibMC::MCAclMessage::MCAclMessage () [inline]

Default constructor. Creates an empty ACL message object.

Definition at line 114 of file MCAclMessage.cs.

References `aclmsg_p`.

Referenced by `Reply()`.

#### 12.72.3.2 internal LibMC::MCAclMessage::MCAclMessage (IntPtr msg) [inline, private]

Definition at line 120 of file MCAclMessage.cs.

References `AclMsg`.

### 12.72.4 Member Function Documentation

#### 12.72.4.1 int LibMC::MCAclMessage::AddReceiver (String name, String address) [inline]

Adds a receiver to the list of receivers. Adds a receiver to the list of receivers for the message.

##### Parameters:

*name* The name of the receiver.  
*address* The address of the receiver.

##### Returns:

The return value of the underlying `MC_AclAddReceiver` function.

##### Note:

The message must be a valid message or this function will fail.

Definition at line 218 of file MCAclMessage.cs.

References `AclMsg`.

**12.72.4.2 int LibMC::MCAclMessage::AddReplyTo (String *name*, String *address*) [inline]**

Adds a "reply-to" field to the message. Adds a "reply-to" field to the message. The reply-to field overrides the sender field when creating a reply.

**Parameters:**

*name* The name of the receiver.  
*address* The address of the receiver.

**Returns:**

The return value of the underlying MC\_AclAddAddReplyTo function.

**Note:**

The message must be a valid message or this function will fail.

Definition at line 235 of file MCAclMessage.cs.

References AclMsg.

**12.72.4.3 int LibMC::MCAclMessage::Destroy () [inline]**

Destroys a message. This function destroys a message in the Mobile-C library. It releases the underlying memory and must be called when the message is no longer needed.

**Returns:**

The return value of the underlying MC\_AclDestroy function.

**Note:**

The message must be a valid message or this function will fail. In addition, messages are not automatically destroyed by the garbage collector. Use care when creating messages and ensure they are properly destroyed.

Definition at line 269 of file MCAclMessage.cs.

References AclMsg, and aclmsg\_p.

**12.72.4.4 void LibMC::MCAclMessage::New () [inline]**

Creates a new, blank ACL message. Creates a new ACL message. The message is blank but valid.

Definition at line 153 of file MCAclMessage.cs.

References AclMsg.

**12.72.4.5 MCAclMessage LibMC::MCAclMessage::Reply (MCAclMessage *acl\_message*) [inline]**

Creates an ACL message that is a response to the argument. Creates an ACL message to respond to the argument.

**Parameters:**

*acl\_message* The message from which to create the reply.

**Returns:**

A new ACL message that is a response to the argument or an empty message if there is an error.

Definition at line 167 of file MCAclMessage.cs.

References AclMsg, and MCAclMessage().

**12.72.4.6 int LibMC::MCAclMessage::SetContent (String *content*) [inline]**

Sets the content field of the message. Sets the content field of the message.

**Parameters:**

*content* The string to copy to the content field.

**Returns:**

The return value of the underlying MC\_AclSetContent function.

**Note:**

The message must be a valid message or this function will fail.

Definition at line 250 of file MCAclMessage.cs.

References AclMsg.

**12.72.4.7 int LibMC::MCAclMessage::SetPerformative (MC\_FipaPerformative\_e *performative*) [inline]**

Sets the performative field of the message. Sets the performative field of the message.

**Parameters:**

*performative* The fipa\_performative\_e enum describing the message.

**Returns:**

The return value of the underlying MC\_AclSetPerformative function.

**Note:**

The message must be a valid message or this function will fail.

Definition at line 186 of file MCAclMessage.cs.

References AclMsg.

#### 12.72.4.8 `int LibMC::MCAclMessage::SetSender (String name, String address) [inline]`

Sets the sender field of the message. Sets the performative field of the message.

##### Parameters:

- name* The name of the sending entity.  
*address* The address of the sending entity.

##### Returns:

The return value of the underlying MC\_AclSetSender function.

##### Note:

The message must be a valid message or this function will fail.

Definition at line 202 of file MCAclMessage.cs.

References AclMsg.

### 12.72.5 Field Documentation

#### 12.72.5.1 `IntPtr LibMC::MCAclMessage::aclmsg_p [private]`

Definition at line 105 of file MCAclMessage.cs.

Referenced by Destroy(), and MCAclMessage().

### 12.72.6 Property Documentation

#### 12.72.6.1 `internal IntPtr LibMC::MCAclMessage::AclMsg [get, set, private]`

Definition at line 129 of file MCAclMessage.cs.

Referenced by LibMC::MCAgent::AclPost(), LibMC::MCAgency::AclSend(), AddReceiver(), AddReplyTo(), Destroy(), MCAclMessage(), New(), Reply(), SetContent(), SetPerformative(), and SetSender().

The documentation for this class was generated from the following file:

- </home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/win32/LibMC.net/LibMC/MCAclMessage.cs>

## 12.73 LibMC::MCAgency Class Reference

Wrapper class for [MCAgency\\_t](#) structure.

### Data Structures

- struct [ChOptions\\_t](#)  
*ChOptions structures.*
- struct [MCAgency\\_t](#)
- struct [MCAgencyOptions\\_t](#)

### Public Types

- enum [MCAgencyState](#) {  
[NoState](#) = -1, [Initialized](#) = 0, [Running](#), [Halted](#),  
[Ended](#) }  
*Enum for describing the state of the agency.*
- enum [ChShellType](#) { [CH\\_REGULARCH](#) = 0, [CH\\_SAFECH](#) = 1 }  
*Ch shell type.*
- enum [MC\\_ThreadIndex\\_e](#) {  
[MC\\_THREAD\\_DF](#) = 0, [MC\\_THREAD\\_AMS](#), [MC\\_THREAD\\_ACC](#), [MC\\_THREAD\\_CP](#),  
[MC\\_THREAD\\_AGENT](#), [MC\\_THREAD\\_ALL](#) }  
*Enum for describing the different threads that Mobile-C uses.*
- enum [MC\\_SteerCommand\\_e](#) { [MC\\_RUN](#) = 0, [MC\\_SUSPEND](#), [MC\\_RESTART](#), [MC\\_STOP](#) }  
*Available commands for MC\_Steer.*
- enum [MC\\_Signal\\_e](#) {  
[MC\\_NO\\_SIGNAL](#) = 0x00, [MC\\_RECV\\_CONNECTION](#) = 0x01, [MC\\_RECV\\_MESSAGE](#) = 0x02,  
[MC\\_RECV\\_AGENT](#) = 0x04,  
[MC\\_RECV\\_RETURN](#) = 0x08, [MC\\_EXEC\\_AGENT](#) = 0x10, [MC\\_ALL\\_SIGNALS](#) = 0x20 }  
*MobileC system signals.*

### Public Member Functions

- [MCAgency](#) ()  
*Default constructor.*
- [int Initialize](#) ()  
*Starts the agency.*
- [int End](#) ()  
*Stops and destroys the agency.*

- [int ChInitializeOptions](#) ([ChShellType](#) shellType, String home)  
*Initializes Ch [options](#) for the agency.*
- [int SetThreadsAllOn](#) ()
- [int SetThreadsAllOff](#) ()  
*Sets all threads for the agency to "off".*
- [int SetThreadOn](#) ([MC\\_ThreadIndex\\_e](#) index)  
*Sets an individual thread for the agency to "on".*
- [int SetThreadOff](#) ([MC\\_ThreadIndex\\_e](#) index)  
*Sets an individual thread for the agency to "off".*
- [int HaltAgency](#) ()  
*Temporarily halts the agency.*
- [int ResumeAgency](#) ()  
*Resumes a halted agency.*
- [int SetDefaultAgentStatus](#) ([MCAgent.MC\\_AgentStatus\\_e](#) status)  
*Sets the default state of an agent in the agency.*
- [MCAgent WaitRetrieveAgent](#) ()  
*Waits for an agent to arrive and returns the agent.*
- [int WaitAgent](#) ()  
*Waits for an agent to arrive.*
- [int SendAgentMigrationMessageFile](#) (String filename, String hostname, [int](#) port)  
*Sends an agent migration message file to an agency.*
- [int LoadAgentMigrationMessageFile](#) (String filename)  
*Load an agent migration message.*
- [int SendAgentMigrationMessage](#) (String message, String hostname, [int](#) port)  
*Sends an agent migration message to an agency.*
- [int CondBroadcast](#) ([int](#) id)  
*Broadcast a condition signal.*
- [int CondSignal](#) ([int](#) id)  
*Signal a condition.*
- [int CondReset](#) ([int](#) id)  
*Reset a condition signal.*
- [int CondWait](#) ([int](#) id)  
*Wait for a condition signal.*



- [int MutexLock \(int id\)](#)  
*Lock a mutex.*
- [int MutexUnlock \(int id\)](#)  
*Unlock a mutex.*
- [int SemaphorePost \(int id\)](#)  
*Posts a semaphore.*
- [int SemaphoreWait \(int id\)](#)  
*Wait for a semaphore to be posted.*
- [int ResetSignal \(\)](#)  
*Reset an agency signal.*
- [int SyncDelete \(int id\)](#)  
*Delete a synchronization variable.*
- [int SyncInit \(int id\)](#)  
*Create a new synchronization variable.*
- [int WaitSignal \(MC\\_Signal\\_esignals\)](#)  
*Wait for agency signals.*
- [int BarrierDelete \(int id\)](#)  
*Delete a barrier object.*
- [int BarrierInit \(int id, int num\\_procs\)](#)  
*Create a new barrier.*
- [MC\\_SteerCommand\\_e SteerControl \(\)](#)  
*Steering control function.*
- [int Steer \(IntPtr funcptr, IntPtr arg\)](#)  
*Steering control function.*
- [int RegisterService \(MCAgent agent, int agentID, String agentName, String\[\] serviceNames, int numServices\)](#)  
*Registers services in the agency.*
- [int SearchForService \(String searchString, IntPtr agentNames, IntPtr serviceNames, IntPtr agentIDs, IntPtr numResults\)](#)  
*Searches for services in the agency.*
- [int AddAgent \(MCAgent agent\)](#)  
*Add an agent to the agency.*
- [MCAgent FindAgentByName \(String name\)](#)  
*Finds an agent by its name.*

- [MCAgent FindAgentByID](#) (int id)  
*Find an agent by its ID.*
- [MCAgent RetrieveAgent](#) ()  
*Retrieve an agent from the agency.*
- [int AclSend](#) ([MCAclMessage](#) acl\_message)  
*Send an ACL message to the agency.*
- [int MainLoop](#) ()  
*Makes the agency wait indefinitely.*

## Properties

- [IntPtr Agency](#) [get, set]
- [int Port](#) [get, set]  
*Accessor for the port number of the agency.*
- [MCAgencyState State](#) [get]  
*Accessor for the agency state.*

## Private Member Functions

- static [IntPtr \\_MC\\_Initialize](#) (int port, ref [MCAgencyOptions\\_t](#) options)
- static [int \\_MC\\_End](#) (IntPtr agency)
- static [int \\_MC\\_ChInitializeOptions](#) (IntPtr agency, [ChOptions\\_t](#) options)
- static [int \\_MC\\_InitializeAgencyOptions](#) (ref [MCAgencyOptions\\_t](#) options)
- static [int \\_MC\\_SetThreadsAllOn](#) (ref [MCAgencyOptions\\_t](#) options)
- static [int \\_MC\\_SetThreadsAllOff](#) (ref [MCAgencyOptions\\_t](#) options)
- static [int \\_MC\\_SetThreadOn](#) (ref [MCAgencyOptions\\_t](#) options, [MC\\_ThreadIndex\\_e](#) index)
- static [int \\_MC\\_SetThreadOff](#) (ref [MCAgencyOptions\\_t](#) options, [MC\\_ThreadIndex\\_e](#) index)
- static [int \\_MC\\_HaltAgency](#) (IntPtr agency)
- static [int \\_MC\\_ResumeAgency](#) (IntPtr agency)
- static [int \\_MC\\_SetDefaultAgentStatus](#) (IntPtr agency, [MCAgent.MC\\_AgentStatus\\_e](#) status)
- static [IntPtr \\_MC\\_WaitRetrieveAgent](#) (IntPtr agency)
- static [int \\_MC\\_WaitAgent](#) (IntPtr agency)
- static [int \\_MC\\_MainLoop](#) (IntPtr agency)
- static [int \\_MC\\_SendAgentMigrationMessageFile](#) (IntPtr agency, String filename, String hostname, int port)
- static [int \\_MC\\_SendAgentMigrationMessage](#) (IntPtr agency, String message, String hostname, int port)
- static [int \\_MC\\_CondBroadcast](#) (IntPtr agency, int id)
- static [int \\_MC\\_CondSignal](#) (IntPtr agency, int id)
- static [int \\_MC\\_CondReset](#) (IntPtr agency, int id)
- static [int \\_MC\\_CondWait](#) (IntPtr agency, int id)
- static [int \\_MC\\_MutexLock](#) (IntPtr agency, int id)
- static [int \\_MC\\_MutexUnlock](#) (IntPtr agency, int id)

- static [int \\_MC\\_SemaphorePost](#) (IntPtr agency, [int](#) id)
- static [int \\_MC\\_SemaphoreWait](#) (IntPtr agency, [int](#) id)
- static [int \\_MC\\_ResetSignal](#) (IntPtr agency)
- static [int \\_MC\\_SyncDelete](#) (IntPtr agency, [int](#) id)
- static [int \\_MC\\_SyncInit](#) (IntPtr agency, [int](#) id)
- static [int \\_MC\\_WaitSignal](#) (IntPtr agency, [int](#) signals)
- static [int \\_MC\\_BarrierDelete](#) (IntPtr agency, [int](#) id)
- static [int \\_MC\\_BarrierInit](#) (IntPtr agency, [int](#) id, [int](#) num\_procs)
- static [MC\\_SteerCommand\\_e \\_MC\\_SteerControl](#) ()
- static [int \\_MC\\_Steer](#) (IntPtr agency, IntPtr funcptr, IntPtr arg)
- static [int \\_MC\\_RegisterService](#) (IntPtr agency, IntPtr agent, [int](#) agentID, String agentName, String[ ] serviceNames, [int](#) numServices)
- static [int \\_MC\\_SearchForService](#) (IntPtr agency, String searchString, IntPtr agentNames, IntPtr serviceNames, IntPtr agentIDs, IntPtr numResults)
- static [int \\_MC\\_AddAgent](#) (IntPtr agency, IntPtr agent)
- static internal [int \\_MC\\_DeleteAgent](#) (IntPtr agent)
- static IntPtr [\\_MC\\_FindAgentByName](#) (IntPtr agency, String name)
- static IntPtr [\\_MC\\_FindAgentByID](#) (IntPtr agency, [int](#) ID)
- static IntPtr [\\_MC\\_RetrieveAgent](#) (IntPtr agency)
- static internal [int \\_MC\\_GetAgentID](#) (IntPtr agent)
- static internal String [\\_MC\\_GetAgentName](#) (IntPtr agent)
- static internal [int \\_MC\\_GetAgentNumTasks](#) (IntPtr agent)
- static internal [MCAgent.MC\\_AgentStatus\\_e \\_MC\\_GetAgentStatus](#) (IntPtr agent)
- static internal [MCAgent.MC\\_AgentType\\_e \\_MC\\_GetAgentType](#) (IntPtr agent)
- static internal String [\\_MC\\_GetAgentXMLString](#) (IntPtr agent)
- static internal [int \\_MC\\_PrintAgentCode](#) (IntPtr agent)
- static internal String [\\_MC\\_RetrieveAgentCode](#) (IntPtr agent)
- static internal [int \\_MC\\_SetAgentStatus](#) (IntPtr agent, [MCAgent.MC\\_AgentStatus\\_e](#) status)
- static internal [int \\_MC\\_TerminateAgent](#) (IntPtr agent)
- static internal [int \\_MC\\_CallAgentFunc](#) (IntPtr agent, String funcName, IntPtr returnVal, IntPtr varg)
- static internal IntPtr [\\_MC\\_GetAgentExecEngine](#) (IntPtr agent)
- static internal [int \\_MC\\_GetAgentReturnData](#) (IntPtr agent, [int](#) task\_num, IntPtr data, IntPtr dim, IntPtr extent)
- static internal IntPtr [\\_MC\\_AclNew](#) ()
- static internal [int \\_MC\\_AclPost](#) (IntPtr agent, IntPtr message)
- static internal IntPtr [\\_MC\\_AclReply](#) (IntPtr acl\_message)
- static internal IntPtr [\\_MC\\_AclRetrieve](#) (IntPtr agent)
- static internal [int \\_MC\\_AclSend](#) (IntPtr agency, IntPtr acl\_message)
- static internal IntPtr [\\_MC\\_AclWaitRetrieve](#) (IntPtr agent)
- static internal [int \\_MC\\_AclSetPerformative](#) (IntPtr acl, [MCAclMessage.MC\\_FipaPerformative\\_e](#) performative)
- static internal [int \\_MC\\_AclSetSender](#) (IntPtr acl, String name, String address)
- static internal [int \\_MC\\_AclAddReceiver](#) (IntPtr acl, String name, String address)
- static internal [int \\_MC\\_AclAddReplyTo](#) (IntPtr acl, String name, String address)
- static internal [int \\_MC\\_AclSetContent](#) (IntPtr acl, String content)
- static internal [int \\_MC\\_AclDestroy](#) (IntPtr acl)

## Private Attributes

- IntPtr [agency\\_p](#) = IntPtr.Zero
- [MCAgencyOptions\\_t options](#)
- int [port](#) = -1
- [MCAgencyState state](#) = MCAgencyState.NoState
- const String [mcdll](#)

### 12.73.1 Detailed Description

Wrapper class for [MCAgency\\_t](#) structure. This class provides an interface to the Mobile-C agency. Member functions for the class are generally overloaded versions of the respective functions in the Mobile-C library. The class maintains a pointer to the Mobile-C agency in unmanaged memory. The pointer is not accessible by the user.

Definition at line 341 of file MCAgency.cs.

### 12.73.2 Member Enumeration Documentation

#### 12.73.2.1 enum LibMC::MCAgency::ChShellType

Ch shell type. Used to set the shell type for the Ch interpreter.

##### Enumerator:

*CH\_REGULARCH* Default, regular shell  
*CH\_SAFECH* Safe shell

Definition at line 363 of file MCAgency.cs.

#### 12.73.2.2 enum LibMC::MCAgency::MC\_Signal\_e

MobileC system signals. Each signal is activated after the corresponding action. i.e. The 'MC\_RECV\_MESSAGE' signal is activated after a message is received.

##### Note:

This enum is pulled directly from the Mobile-C library.

##### See also:

[MC\\_WaitSignal\(\)](#), [MC\\_ResetSignal\(\)](#)

##### Enumerator:

*MC\_NO\_SIGNAL*  
*MC\_RECV\_CONNECTION*  
*MC\_RECV\_MESSAGE*  
*MC\_RECV\_AGENT*  
*MC\_RECV\_RETURN*  
*MC\_EXEC\_AGENT*  
*MC\_ALL\_SIGNALS*

Definition at line 112 of file MCEExports.cs.

### 12.73.2.3 enum LibMC::MCAgency::MC\_SteerCommand\_e

Available commands for MC\_Steer.

**Note:**

This enum is pulled directly from the Mobile-C library.

**Enumerator:**

*MC\_RUN* Continue the algorithm  
*MC\_SUSPEND* Suspend/pause the algorithm  
*MC\_RESTART* Restart the algorithm from the beginning  
*MC\_STOP* Stop the algorithm

Definition at line 94 of file MCEExports.cs.

### 12.73.2.4 enum LibMC::MCAgency::MC\_ThreadIndex\_e

Enum for describing the different threads that Mobile-C uses. These enums can be used to turn threads on and off before an agency is initialized.

**Note:**

This enum is pulled directly from the Mobile-C library.

**Enumerator:**

*MC\_THREAD\_DF* Directory Facilitator  
*MC\_THREAD\_AMS* Agent Management system  
*MC\_THREAD\_ACC* Agency communications  
*MC\_THREAD\_CP* Command Prompt  
*MC\_THREAD\_AGENT* Agent threads  
*MC\_THREAD\_ALL*

Definition at line 79 of file MCEExports.cs.

### 12.73.2.5 enum LibMC::MCAgency::MCAgencyState

Enum for describing the state of the agency. This enum is used to determine whether or not certain actions should be permitted, such as halting, resuming, and ending an agency

**Enumerator:**

*NoState* Default, uninitialized state  
*Initialized* Agency initialized, but not started  
*Running* Agency is running  
*Halted* Agency has been stopped (can be resumed)  
*Ended* Agency is stopped (destroyed)

Definition at line 349 of file MCAgency.cs.

### 12.73.3 Constructor & Destructor Documentation

#### 12.73.3.1 LibMC::MCAgency::MCAgency () [inline]

Default constructor. The default constructor for the [MCAgency](#) class. It creates a new agency, default [options](#) for the agency, and initializes the agency. It does not start the agency.

Definition at line 381 of file MCAgency.cs.

References `_MC_InitializeAgencyOptions()`, and `state`.

### 12.73.4 Member Function Documentation

#### 12.73.4.1 static internal int LibMC::MCAgency::\_MC\_AclAddReceiver (IntPtr *acl*, String *name*, String *address*) [private]

#### 12.73.4.2 static internal int LibMC::MCAgency::\_MC\_AclAddReplyTo (IntPtr *acl*, String *name*, String *address*) [private]

#### 12.73.4.3 static internal int LibMC::MCAgency::\_MC\_AclDestroy (IntPtr *acl*) [private]

#### 12.73.4.4 static internal IntPtr LibMC::MCAgency::\_MC\_AclNew () [private]

#### 12.73.4.5 static internal int LibMC::MCAgency::\_MC\_AclPost (IntPtr *agent*, IntPtr *message*) [private]

#### 12.73.4.6 static internal IntPtr LibMC::MCAgency::\_MC\_AclReply (IntPtr *acl\_message*) [private]

#### 12.73.4.7 static internal IntPtr LibMC::MCAgency::\_MC\_AclRetrieve (IntPtr *agent*) [private]

#### 12.73.4.8 static internal int LibMC::MCAgency::\_MC\_AclSend (IntPtr *agency*, IntPtr *acl\_message*) [private]

#### 12.73.4.9 static internal int LibMC::MCAgency::\_MC\_AclSetContent (IntPtr *acl*, String *content*) [private]

#### 12.73.4.10 static internal int LibMC::MCAgency::\_MC\_AclSetPerformative (IntPtr *acl*, MCAclMessage.MC\_FipaPerformative *e\_performative*) [private]

#### 12.73.4.11 static internal int LibMC::MCAgency::\_MC\_AclSetSender (IntPtr *acl*, String *name*, String *address*) [private]

#### 12.73.4.12 static internal IntPtr LibMC::MCAgency::\_MC\_AclWaitRetrieve (IntPtr *agent*) [private]

#### 12.73.4.13 static int LibMC::MCAgency::\_MC\_AddAgent (IntPtr *agency*, IntPtr *agent*) [private]

Referenced by `AddAgent()`.

**12.73.4.14** `static int LibMC::MCAgency::_MC_BarrierDelete (IntPtr agency, int id)  
[private]`

Referenced by BarrierDelete().

**12.73.4.15** `static int LibMC::MCAgency::_MC_BarrierInit (IntPtr agency, int id, int num_procs)  
[private]`

Referenced by BarrierInit().

**12.73.4.16** `static internal int LibMC::MCAgency::_MC_CallAgentFunc (IntPtr agent, String  
funcName, IntPtr returnVal, IntPtr varg) [private]`

**12.73.4.17** `static int LibMC::MCAgency::_MC_ChInitializeOptions (IntPtr agency, ChOptions_t  
options) [private]`

Referenced by ChInitializeOptions().

**12.73.4.18** `static int LibMC::MCAgency::_MC_CondBroadcast (IntPtr agency, int id)  
[private]`

Referenced by CondBroadcast().

**12.73.4.19** `static int LibMC::MCAgency::_MC_CondReset (IntPtr agency, int id) [private]`

Referenced by CondReset().

**12.73.4.20** `static int LibMC::MCAgency::_MC_CondSignal (IntPtr agency, int id) [private]`

Referenced by CondSignal().

**12.73.4.21** `static int LibMC::MCAgency::_MC_CondWait (IntPtr agency, int id) [private]`

Referenced by CondWait().

**12.73.4.22** `static internal int LibMC::MCAgency::_MC_DeleteAgent (IntPtr agent) [private]`

**12.73.4.23** `static int LibMC::MCAgency::_MC_End (IntPtr agency) [private]`

**12.73.4.24** `static IntPtr LibMC::MCAgency::_MC_FindAgentByID (IntPtr agency, int ID)  
[private]`

Referenced by FindAgentByID().

**12.73.4.25** `static IntPtr LibMC::MCAgency::_MC_FindAgentByName (IntPtr agency, String  
name) [private]`

Referenced by FindAgentByName().

- 12.73.4.26 **static internal IntPtr LibMC::MCAgency::\_MC\_GetAgentExecEngine (IntPtr *agent*)**  
[private]
- 12.73.4.27 **static internal int LibMC::MCAgency::\_MC\_GetAgentID (IntPtr *agent*)** [private]
- 12.73.4.28 **static internal String LibMC::MCAgency::\_MC\_GetAgentName (IntPtr *agent*)**  
[private]
- 12.73.4.29 **static internal int LibMC::MCAgency::\_MC\_GetAgentNumTasks (IntPtr *agent*)**  
[private]
- 12.73.4.30 **static internal int LibMC::MCAgency::\_MC\_GetAgentReturnData (IntPtr *agent*, int**  
***task\_num*, IntPtr *data*, IntPtr *dim*, IntPtr *extent*)** [private]
- 12.73.4.31 **static internal MCAgent.MC\_AgentStatus\_e LibMC::MCAgency::\_MC\_-**  
**GetAgentStatus (IntPtr *agent*)** [private]
- 12.73.4.32 **static internal MCAgent.MC\_AgentType\_e LibMC::MCAgency::\_MC\_GetAgentType**  
**(IntPtr *agent*)** [private]
- 12.73.4.33 **static internal String LibMC::MCAgency::\_MC\_GetAgentXMLString (IntPtr *agent*)**  
[private]
- 12.73.4.34 **static int LibMC::MCAgency::\_MC\_HaltAgency (IntPtr *agency*)** [private]

Referenced by HaltAgency().

- 12.73.4.35 **static IntPtr LibMC::MCAgency::\_MC\_Initialize (int *port*, ref MCAgencyOptions\_t**  
***options*)** [private]
- 12.73.4.36 **static int LibMC::MCAgency::\_MC\_InitializeAgencyOptions (ref**  
**MCAgencyOptions\_t *options*)** [private]

Referenced by MCAgency().

- 12.73.4.37 **static int LibMC::MCAgency::\_MC\_MainLoop (IntPtr *agency*)** [private]

Referenced by MainLoop().

- 12.73.4.38 **static int LibMC::MCAgency::\_MC\_MutexLock (IntPtr *agency*, int *id*)** [private]

Referenced by MutexLock().

- 12.73.4.39 **static int LibMC::MCAgency::\_MC\_MutexUnlock (IntPtr *agency*, int *id*)**  
[private]

Referenced by MutexUnlock().



**12.73.4.40** `static internal int LibMC::MCAgency::_MC_PrintAgentCode (IntPtr agent)  
[private]`

**12.73.4.41** `static int LibMC::MCAgency::_MC_RegisterService (IntPtr agency, IntPtr agent, int  
agentID, String agentName, String[] serviceNames, int numServices) [private]`

Referenced by RegisterService().

**12.73.4.42** `static int LibMC::MCAgency::_MC_ResetSignal (IntPtr agency) [private]`

Referenced by ResetSignal().

**12.73.4.43** `static int LibMC::MCAgency::_MC_ResumeAgency (IntPtr agency) [private]`

Referenced by ResumeAgency().

**12.73.4.44** `static IntPtr LibMC::MCAgency::_MC_RetrieveAgent (IntPtr agency) [private]`

Referenced by RetrieveAgent().

**12.73.4.45** `static internal String LibMC::MCAgency::_MC_RetrieveAgentCode (IntPtr agent)  
[private]`

**12.73.4.46** `static int LibMC::MCAgency::_MC_SearchForService (IntPtr agency, String  
searchString, IntPtr agentNames, IntPtr serviceNames, IntPtr agentIDs, IntPtr  
numResults) [private]`

**12.73.4.47** `static int LibMC::MCAgency::_MC_SemaphorePost (IntPtr agency, int id)  
[private]`

Referenced by SemaphorePost().

**12.73.4.48** `static int LibMC::MCAgency::_MC_SemaphoreWait (IntPtr agency, int id)  
[private]`

Referenced by SemaphoreWait().

**12.73.4.49** `static int LibMC::MCAgency::_MC_SendAgentMigrationMessage (IntPtr agency,  
String message, String hostname, int port) [private]`

Referenced by SendAgentMigrationMessage().

**12.73.4.50** `static int LibMC::MCAgency::_MC_SendAgentMigrationMessageFile (IntPtr agency, String filename, String hostname, int port) [private]`

**12.73.4.51** `static internal int LibMC::MCAgency::_MC_SetAgentStatus (IntPtr agent, MCAgent.MC_AgentStatus_e status) [private]`

**12.73.4.52** `static int LibMC::MCAgency::_MC_SetDefaultAgentStatus (IntPtr agency, MCAgent.MC_AgentStatus_e status) [private]`

Referenced by SetDefaultAgentStatus().

**12.73.4.53** `static int LibMC::MCAgency::_MC_SetThreadOff (ref MCAgencyOptions_t options, MC_ThreadIndex_e index) [private]`

Referenced by SetThreadOff().

**12.73.4.54** `static int LibMC::MCAgency::_MC_SetThreadOn (ref MCAgencyOptions_t options, MC_ThreadIndex_e index) [private]`

Referenced by SetThreadOn().

**12.73.4.55** `static int LibMC::MCAgency::_MC_SetThreadsAllOff (ref MCAgencyOptions_t options) [private]`

Referenced by SetThreadsAllOff().

**12.73.4.56** `static int LibMC::MCAgency::_MC_SetThreadsAllOn (ref MCAgencyOptions_t options) [private]`

Referenced by SetThreadsAllOn().

**12.73.4.57** `static int LibMC::MCAgency::_MC_Steer (IntPtr agency, IntPtr funcptr, IntPtr arg) [private]`

**12.73.4.58** `static MC_SteerCommand_e LibMC::MCAgency::_MC_SteerControl () [private]`

Referenced by SteerControl().

**12.73.4.59** `static int LibMC::MCAgency::_MC_SyncDelete (IntPtr agency, int id) [private]`

Referenced by SyncDelete().

**12.73.4.60** `static int LibMC::MCAgency::_MC_SyncInit (IntPtr agency, int id) [private]`

Referenced by SyncInit().

**12.73.4.61** `static internal int LibMC::MCAgency::_MC_TerminateAgent (IntPtr agent)`  
[private]

**12.73.4.62** `static int LibMC::MCAgency::_MC_WaitAgent (IntPtr agency)` [private]

Referenced by WaitAgent().

**12.73.4.63** `static IntPtr LibMC::MCAgency::_MC_WaitRetrieveAgent (IntPtr agency)`  
[private]

Referenced by WaitRetrieveAgent().

**12.73.4.64** `static int LibMC::MCAgency::_MC_WaitSignal (IntPtr agency, int signals)`  
[private]

Referenced by WaitSignal().

**12.73.4.65** `int LibMC::MCAgency::AclSend (MCAclMessage acl_message)` [inline]

Send an ACL message to the agency. Sends an ACL message to the agency. The message is delivered appropriately.

**Parameters:**

*acl\_message* The message to send.

**Returns:**

The return value of the underlying MC\_CondBroadcast function.

Definition at line 1048 of file MCAgency.cs.

References LibMC::MCAclMessage::AclMsg, and Agency.

**12.73.4.66** `int LibMC::MCAgency::AddAgent (MCAgent agent)` [inline]

Add an agent to the agency. Adds an agent to the agency.

**Parameters:**

*agent* The agent to add.

**Returns:**

The return value of the underlying MC\_AddAgent function.

Definition at line 993 of file MCAgency.cs.

References \_MC\_AddAgent(), Agency, and LibMC::MCAgent::Agent.

**12.73.4.67 int LibMC::MCAgency::BarrierDelete (int *id*) [inline]**

Delete a barrier object. Deletes a barrier object from the agency. The parameter "id" is the ID of the agency sync variable created with [BarrierInit\(\)](#).

**Parameters:**

*id* The ID number of the barrier to delete.

**Returns:**

The return value of the underlying MC\_BarrierDelete function.

Definition at line 877 of file MCAgency.cs.

References `_MC_BarrierDelete()`, and `Agency`.

**12.73.4.68 int LibMC::MCAgency::BarrierInit (int *id*, int *num\_procs*) [inline]**

Create a new barrier. Creates a new barrier object in the agency.

**Parameters:**

*id* The ID number of the condition to signal.

*num\_procs* the number of process to block (?)

**Returns:**

The return value of the underlying MC\_BarrierInit function.

Definition at line 891 of file MCAgency.cs.

References `_MC_BarrierInit()`, and `Agency`.

**12.73.4.69 int LibMC::MCAgency::ChInitializeOptions (ChShellType *shellType*, String *home*) [inline]**

Initializes Ch [options](#) for the agency. Can be used to set the home directory and shell mode for the Ch interpreter.

**Parameters:**

*shellType* The type of shell Ch should use: CH\_REGULARARCH or CH\_SAFECH.

*home* The home directory Ch should use.

**Returns:**

The return value of the underlying MC\_ChInitializeOptions function.

**Note:**

This function must be called before the agency is started.

Definition at line 499 of file MCAgency.cs.

References `_MC_ChInitializeOptions()`, `Agency`, `LibMC::MCAgency::ChOptions_t::chhome`, `int`, and `LibMC::MCAgency::ChOptions_t::shelltype`.

**12.73.4.70 int LibMC::MCAgency::CondBroadcast (int *id*) [inline]**

Broadcast a condition signal. Broadcasts a signal in the agency. The parameter "id" is the ID of the agency sync variable created with [SyncInit\(\)](#).

**Parameters:**

*id* The ID number of the condition to signal.

**Returns:**

The return value of the underlying MC\_CondBroadcast function.

Definition at line 705 of file MCAgency.cs.

References [\\_MC\\_CondBroadcast\(\)](#), and [Agency](#).

**12.73.4.71 int LibMC::MCAgency::CondReset (int *id*) [inline]**

Reset a condition signal. Resets a signal in the agency. The parameter "id" is the ID of the agency sync variable created with [SyncInit\(\)](#). This function must be called after a condition is received in order to clear it.

**Parameters:**

*id* The ID number of the condition to reset.

**Returns:**

The return value of the underlying MC\_CondReset function.

Definition at line 734 of file MCAgency.cs.

References [\\_MC\\_CondReset\(\)](#), and [Agency](#).

**12.73.4.72 int LibMC::MCAgency::CondSignal (int *id*) [inline]**

Signal a condition. Signals a condition in the agency. The parameter "id" is the ID of the agency sync variable to signal that was created with [SyncInit\(\)](#).

**Parameters:**

*id* The ID number of the condition to signal.

**Returns:**

The return value of the underlying MC\_CondSignal function.

Definition at line 719 of file MCAgency.cs.

References [\\_MC\\_CondSignal\(\)](#), and [Agency](#).

**12.73.4.73 int LibMC::MCAgency::CondWait (int *id*) [inline]**

Wait for a condition signal. Waits for a condition signal in the agency. The parameter "id" is the ID of the agency sync variable created with [SyncInit\(\)](#). This function blocks until the signal is received.

**Parameters:**

*id* The ID number of the condition to wait for.

**Returns:**

The return value of the underlying MC\_CondWait function.

Definition at line 749 of file MCAgency.cs.

References \_MC\_CondWait(), and Agency.

**12.73.4.74 int LibMC::MCAgency::End () [inline]**

Stops and destroys the agency. Stops the agency and sets the agency state appropriately.

**Returns:**

The return value of the underlying MC\_End function.

**Note:**

This call will fail if the underlying Mobile-C agency is not in the correct state.

Definition at line 481 of file MCAgency.cs.

References Agency, and state.

**12.73.4.75 MCAgent LibMC::MCAgency::FindAgentByID (int *id*) [inline]**

Find an agent by its ID. Finds an agent in the agency by its ID number.

**Parameters:**

*id* The ID number of the condition to signal.

**Returns:**

The return value of the underlying MC\_FindAgentByID function.

Definition at line 1019 of file MCAgency.cs.

References \_MC\_FindAgentByID(), and Agency.

**12.73.4.76 MCAgent LibMC::MCAgency::FindAgentByName (String *name*) [inline]**

Finds an agent by its name. Finds an agent in the agency by its name.

**Parameters:**

*name* The name of the agent to search for.

**Returns:**

The return value of the underlying MC\_FindAgentByName function.

Definition at line 1006 of file MCAgency.cs.

References \_MC\_FindAgentByName(), and Agency.

**12.73.4.77 int LibMC::MCAgency::HaltAgency () [inline]**

Temporarily halts the agency. Halts the agency until it is resumed or ended.

**Returns:**

The return value of the underlying MC\_HaltAgency function.

**Note:**

The underlying Mobile-C agency must be in the correct state to call this function or it will fail.

Definition at line 576 of file MCAgency.cs.

References `_MC_HaltAgency()`, `Agency`, and `state`.

**12.73.4.78 int LibMC::MCAgency::Initialize () [inline]**

Starts the agency. Starts the agency and sets the agency state.

**Returns:**

0 on success, -1 on failure.

**Note:**

The agency port and any other [options](#) must be set before calling this function.

Definition at line 459 of file MCAgency.cs.

References `Agency`, `port`, and `state`.

**12.73.4.79 int LibMC::MCAgency::LoadAgentMigrationMessageFile (String filename) [inline]**

Load an agent migration message. Loads the specified XML file to this agency automatically. There is no need to specify a port or agency location.

**Parameters:**

*filename* The name of the file to send (fully qualified).

**Returns:**

The return value of the underlying MC\_SendAgentMigrationMessageFile function.

Definition at line 672 of file MCAgency.cs.

References `Agency`, and `Port`.

**12.73.4.80 int LibMC::MCAgency::MainLoop () [inline]**

Makes the agency wait indefinitely. Makes the agency wait indefinitely until it receives a "quit" command or is otherwise terminated.

**Returns:**

The return value of the underlying MC\_MainLoop function.

Definition at line 1061 of file MCAgency.cs.

References \_MC\_MainLoop(), and Agency.

**12.73.4.81 int LibMC::MCAgency::MutexLock (int *id*) [inline]**

Lock a mutex. Locks a mutex in the agency. The parameter "id" is the ID of the agency sync variable created with [SyncInit\(\)](#). This function blocks until the mutex is locked.

**Parameters:**

*id* The ID number of the mutex to lock.

**Returns:**

The return value of the underlying MC\_MutexLock function.

Definition at line 764 of file MCAgency.cs.

References \_MC\_MutexLock(), and Agency.

**12.73.4.82 int LibMC::MCAgency::MutexUnlock (int *id*) [inline]**

Unlock a mutex. Locks a mutex in the agency. The parameter "id" is the ID of the agency sync variable created with [SyncInit\(\)](#).

**Parameters:**

*id* The ID number of the mutex to unlock.

**Returns:**

The return value of the underlying MC\_MutexUnlock function.

Definition at line 778 of file MCAgency.cs.

References \_MC\_MutexUnlock(), and Agency.

**12.73.4.83 int LibMC::MCAgency::RegisterService (MCAgent *agent*, int *agentID*, String *agentName*, String[] *serviceNames*, int *numServices*) [inline]**

Registers services in the agency. Registers services provided by agents with the agency. Not really useful in binary space.

**Parameters:**

*agent* The agent providing the services.

*agentID* The agent ID number.

*agentName* The agent name.

*serviceNames* An array of service names.



*numServices* The number of services provided.

**Returns:**

The return value of the underlying MC\_RegisterService function.

**Todo**

Test MC\_RegisterService and MC\_SearchForService.

Definition at line 952 of file MCAgency.cs.

References `_MC_RegisterService()`, `Agency`, and `LibMC::MCAgent::Agent`.

**12.73.4.84 int LibMC::MCAgency::ResetSignal () [inline]**

Reset an agency signal. Resets a signal in the agency. The parameter "id" is the ID of the agency sync variable created with [SyncInit\(\)](#).

**Returns:**

The return value of the underlying MC\_ResetSignal function.

Definition at line 820 of file MCAgency.cs.

References `_MC_ResetSignal()`, and `Agency`.

**12.73.4.85 int LibMC::MCAgency::ResumeAgency () [inline]**

Resumes a halted agency. Resumes a halted agency. Cannot be used on ended agencies.

**Returns:**

The return value of the underlying MC\_ResumeAgency function.

**Note:**

The underlying Mobile-C agency must be in the correct state to call this function or it will fail.

Definition at line 593 of file MCAgency.cs.

References `_MC_ResumeAgency()`, `Agency`, and `state`.

**12.73.4.86 MCAgent LibMC::MCAgency::RetrieveAgent () [inline]**

Retrieve an agent from the agency. Really not sure.

**Returns:**

The return value of the underlying MC\_CondBroadcast function.

Definition at line 1031 of file MCAgency.cs.

References `_MC_RetrieveAgent()`, and `Agency`.

### 12.73.4.87 `int LibMC::MCAgency::SearchForService (String searchString, IntPtr agentNames, IntPtr serviceNames, IntPtr agentIDs, IntPtr numResults) [inline]`

Searches for services in the agency. Searches for services provided by agents with the agency. Not really useful in binary space.

#### Parameters:

*searchString* The agent providing the services.

*agentNames* The agent ID number.

*serviceNames* The agent name.

*agentIDs* An array of service names.

*numResults* The number of services provided.

#### Returns:

The return value of the underlying MC\_SearchForService function.

#### Note:

This function does nothing but throw an exception right now.

#### Bug

MC\_SearchForService is not yet implemented.

#### Todo

Implement SearchForService

Definition at line 975 of file MCAgency.cs.

### 12.73.4.88 `int LibMC::MCAgency::SemaphorePost (int id) [inline]`

Posts a semaphore. Posts a semaphore in the agency. The parameter "id" is the ID of the agency sync variable created with [SyncInit\(\)](#).

#### Parameters:

*id* The ID number of the semaphore to post.

#### Returns:

The return value of the underlying MC\_SemaphorePost function.

Definition at line 792 of file MCAgency.cs.

References `_MC_SemaphorePost()`, and `Agency`.

### 12.73.4.89 `int LibMC::MCAgency::SemaphoreWait (int id) [inline]`

Wait for a semaphore to be posted. Wait for a semaphore in the agency to be posted. The parameter "id" is the ID of the agency sync variable created with [SyncInit\(\)](#). This function blocks until the semaphore is posted.

**Parameters:**

*id* The ID number of the semaphore to wait for.

**Returns:**

The return value of the underlying MC\_SemaphoreWait function.

Definition at line 807 of file MCAgency.cs.

References \_MC\_SemaphoreWait(), and Agency.

**12.73.4.90 int LibMC::MCAgency::SendAgentMigrationMessage (String message, String hostname, int port) [inline]**

Sends an agent migration message to an agency. Sends an agent migration message to another agency (local or remote).

**Parameters:**

*message* The agent migration message.

*hostname* The URL, IP address, or other identifier for the agency host.

*port* The port to send to.

**Returns:**

The return value of the underlying MC\_SendAgentMigrationMessageFile function.

Definition at line 687 of file MCAgency.cs.

References \_MC\_SendAgentMigrationMessage(), and Agency.

**12.73.4.91 int LibMC::MCAgency::SendAgentMigrationMessageFile (String filename, String hostname, int port) [inline]**

Sends an agent migration message file to an agency. Sends the specified XML file to another agency (local or remote).

**Parameters:**

*filename* The name of the file to send (fully qualified).

*hostname* The URL, IP address, or other identifier for the agency host.

*port* The port to send to.

**Returns:**

The return value of the underlying MC\_SendAgentMigrationMessageFile function.

Definition at line 658 of file MCAgency.cs.

References Agency.

**12.73.4.92 int LibMC::MCAgency::SetDefaultAgentStatus (MCAgent.MC\_AgentStatus\_e *status*) [inline]**

Sets the default state of an agent in the agency. Can be used to set the default status of agents, but most agents managed their state on their own.

**Parameters:**

*status* The enum that identifies the desired agent state.

**Returns:**

The return value of the underlying MC\_SetDefaultAgentStatus function.

Definition at line 609 of file MCAgency.cs.

References \_MC\_SetDefaultAgentStatus(), and Agency.

**12.73.4.93 int LibMC::MCAgency::SetThreadOff (MC\_ThreadIndex\_e *index*) [inline]**

Sets an individual thread for the agency to "off.". Most commonly used to turn the command prompt thread off.

**Parameters:**

*index* The enum that identifies the thread to be turned off.

**Returns:**

The return value of the underlying MC\_SetThreadOff function.

**Note:**

This function must be called before the agency is started.

Definition at line 561 of file MCAgency.cs.

References \_MC\_SetThreadOff().

**12.73.4.94 int LibMC::MCAgency::SetThreadOn (MC\_ThreadIndex\_e *index*) [inline]**

Sets an individual thread for the agency to "on.". Threads are on by default. If they have been turned off, this function turns them on again.

**Parameters:**

*index* The enum that identifies the thread to be turned on.

**Returns:**

The return value of the underlying MC\_SetThreadOn function.

**Note:**

This function must be called before the agency is started.

Definition at line 546 of file MCAgency.cs.

References \_MC\_SetThreadOn().

**12.73.4.95 int LibMC::MCAgency::SetThreadsAllOff () [inline]**

Sets all threads for the agency to "off.". Sets all threads for the agency to "off." Not recommended for use.

**Returns:**

The return value of the underlying MC\_SetThreadsAllOff function.

**Note:**

This function must be called before the agency is started.

Definition at line 530 of file MCAgency.cs.

References `_MC_SetThreadsAllOff()`.

**12.73.4.96 int LibMC::MCAgency::SetThreadsAllOn () [inline]**

Definition at line 516 of file MCAgency.cs.

References `_MC_SetThreadsAllOn()`.

**12.73.4.97 int LibMC::MCAgency::Steer (IntPtr funcptr, IntPtr arg) [inline]**

Steering control function. Really not sure.

**Parameters:**

*funcptr* Pointer to the steering function

*arg* Argument to function

**Returns:**

The return value of the underlying `_MC_Steer` function.

**Note:**

This function does nothing but throw an exception right now.

**Bug**

`MC_Steer` is not yet implemented.

**Todo**

Implement `MC_Steer`

Definition at line 927 of file MCAgency.cs.

**12.73.4.98 MC\_SteerCommand\_e LibMC::MCAgency::SteerControl () [inline]**

Steering control function. Really not sure.

**Returns:**

The return value of the underlying `MC_SteerControl` function.

**Todo**

Test MC\_SteerControl, MC\_Steer.

Definition at line 909 of file MCAgency.cs.

References \_MC\_SteerControl().

**12.73.4.99 int LibMC::MCAgency::SyncDelete (int *id*) [inline]**

Delete a synchronization variable. Deletes a synchronization variable in the agency. The parameter "id" is the ID of the agency sync variable created with [SyncInit\(\)](#).

**Parameters:**

*id* The ID number of the variable to delete.

**Returns:**

The return value of the underlying MC\_SyncDelete function.

Definition at line 834 of file MCAgency.cs.

References \_MC\_SyncDelete(), and Agency.

**12.73.4.100 int LibMC::MCAgency::SyncInit (int *id*) [inline]**

Create a new synchronization variable. Creates a new synchronization variable in the agency. The parameter "id" is desired ID of the variable. A random ID is returned if "id" is already in use.

**Parameters:**

*id* The ID number of the condition to signal.

**Returns:**

The return value of the underlying MC\_CondBroadcast function- either a random ID or the desired ID if the desired ID is already in use.

Definition at line 850 of file MCAgency.cs.

References \_MC\_SyncInit(), and Agency.

**12.73.4.101 int LibMC::MCAgency::WaitAgent () [inline]**

Waits for an agent to arrive. Waits for an agent to arrive in the agency. The agent is allowed to execute normally.

**Returns:**

The return value of the underlying MC\_WaitAgent function.

Definition at line 639 of file MCAgency.cs.

References \_MC\_WaitAgent(), and Agency.

**12.73.4.102 MCAgent LibMC::MCAgency::WaitRetrieveAgent () [inline]**

Waits for an agent to arrive and returns the agent. Waits for an agent to arrive in the agency, then returns that agent. The agent is not allowed to execute.

**Returns:**

The agent that was retrieved or an empty agent if it fails.

Definition at line 622 of file MCAgency.cs.

References `_MC_WaitRetrieveAgent()`, and `Agency`.

**12.73.4.103 int LibMC::MCAgency::WaitSignal (MC\_Signal\_e signals) [inline]**

Wait for agency signals. Waits for signals to occur in the agency.

**Parameters:**

*signals* The ID number of the condition to signal.

**Returns:**

The return value of the underlying `MC_WaitSignal` function.

Definition at line 863 of file MCAgency.cs.

References `_MC_WaitSignal()`, and `Agency`.

**12.73.5 Field Documentation****12.73.5.1 IntPtr LibMC::MCAgency::agency\_p = IntPtr.Zero [private]**

Definition at line 369 of file MCAgency.cs.

**12.73.5.2 const String LibMC::MCAgency::mcdll [private]****Initial value:**

```
"libmc.dll"
```

Definition at line 15 of file MCEExports.cs.

**12.73.5.3 MCAgencyOptions\_t LibMC::MCAgency::options [private]**

Definition at line 370 of file MCAgency.cs.

#### 12.73.5.4 `int LibMC::MCAgency::port = -1` `[private]`

Definition at line 371 of file MCAgency.cs.

Referenced by `Initialize()`.

#### 12.73.5.5 `MCAgencyState LibMC::MCAgency::state = MCAgencyState.NoState` `[private]`

Definition at line 372 of file MCAgency.cs.

Referenced by `End()`, `HaltAgency()`, `Initialize()`, `MCAgency()`, and `ResumeAgency()`.

### 12.73.6 Property Documentation

#### 12.73.6.1 `IntPtr LibMC::MCAgency::Agency` `[get, set, private]`

Definition at line 389 of file MCAgency.cs.

Referenced by `AclSend()`, `AddAgent()`, `BarrierDelete()`, `BarrierInit()`, `ChInitializeOptions()`, `CondBroadcast()`, `CondReset()`, `CondSignal()`, `CondWait()`, `End()`, `FindAgentByID()`, `FindAgentByName()`, `HaltAgency()`, `Initialize()`, `LoadAgentMigrationMessageFile()`, `MainLoop()`, `MutexLock()`, `MutexUnlock()`, `RegisterService()`, `ResetSignal()`, `ResumeAgency()`, `RetrieveAgent()`, `SemaphorePost()`, `SemaphoreWait()`, `SendAgentMigrationMessage()`, `SendAgentMigrationMessageFile()`, `SetDefaultAgentStatus()`, `SyncDelete()`, `SyncInit()`, `WaitAgent()`, `WaitRetrieveAgent()`, and `WaitSignal()`.

#### 12.73.6.2 `int LibMC::MCAgency::Port` `[get, set]`

Accessor for the port number of the agency. Allows the user to set the agency port or get the port number while it is running.

##### **Note:**

The port must be set before the agency is started. Once the agency is started, the port cannot be changed.

Definition at line 418 of file MCAgency.cs.

Referenced by `LoadAgentMigrationMessageFile()`.

#### 12.73.6.3 `MCAgencyState LibMC::MCAgency::State` `[get]`

Accessor for the agency state. Allows the user to query the state of the agency.

##### **Note:**

The state cannot be set by the user. It is controlled internally.

Definition at line 438 of file MCAgency.cs.

The documentation for this class was generated from the following files:

- </home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/win32/LibMC.net/LibMC/MCAgency.cs>
- </home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/win32/LibMC.net/LibMC/MCExports.cs>



## 12.74 LibMC::MCAgency::MCAgency\_t Struct Reference

### Data Fields

- [int client](#)
- [int server](#)
- [String hostName](#)
- [String filename](#)
- [int portno](#)
- [int portnoc](#)
- [IntPtr mc\\_platform](#)
- [int default\\_agentstatus](#)
- [int threads](#)
- [int enable\\_security](#)
- [int\[ \] stack\\_size](#)
- [int last\\_error](#)

### 12.74.1 Detailed Description

Definition at line 23 of file MCExports.cs.

### 12.74.2 Field Documentation

#### 12.74.2.1 `int LibMC::MCAgency::MCAgency_t::client`

Definition at line 25 of file MCExports.cs.

#### 12.74.2.2 `int LibMC::MCAgency::MCAgency_t::default_agentstatus`

Agency default agent status

Definition at line 32 of file MCExports.cs.

#### 12.74.2.3 `int LibMC::MCAgency::MCAgency_t::enable_security`

Security flag

Definition at line 34 of file MCExports.cs.

#### 12.74.2.4 `String LibMC::MCAgency::MCAgency_t::filename`

Definition at line 28 of file MCExports.cs.

#### 12.74.2.5 `String LibMC::MCAgency::MCAgency_t::hostName`

Local Hostname

Definition at line 27 of file MCExports.cs.

**12.74.2.6 int LibMC::MCAgency::MCAgency\_t::last\_error**

Definition at line 38 of file MCEExports.cs.

**12.74.2.7 IntPtr LibMC::MCAgency::MCAgency\_t::mc\_platform**

Local MobileC Platform

Definition at line 31 of file MCEExports.cs.

**12.74.2.8 int LibMC::MCAgency::MCAgency\_t::portno**

Local port number

Definition at line 29 of file MCEExports.cs.

**12.74.2.9 int LibMC::MCAgency::MCAgency\_t::portnoc**

Definition at line 30 of file MCEExports.cs.

**12.74.2.10 int LibMC::MCAgency::MCAgency\_t::server**

Definition at line 26 of file MCEExports.cs.

**12.74.2.11 int [ ] LibMC::MCAgency::MCAgency\_t::stack\_size**

Definition at line 37 of file MCEExports.cs.

**12.74.2.12 int LibMC::MCAgency::MCAgency\_t::threads**

flag which determines which threads to start

Definition at line 33 of file MCEExports.cs.

The documentation for this struct was generated from the following file:

- </home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/win32/LibMC.net/LibMC/MCEExports.cs>

## 12.75 MCAgencyOptions\_s Struct Reference

User modifiable agency [options](#).

```
#include <libmc.h>
```

### Data Fields

- [int threads](#)
- [int default\\_agent\\_status](#)
- [int modified](#)
- [int enable\\_security](#)
- unsigned char [passphrase](#) [32]
- [int stack\\_size](#) [MC\_THREAD\_ALL]
- char \* [known\\_host\\_filename](#)
- char \* [priv\\_key\\_filename](#)
- [int initInterps](#)
- ChOptions\_t \* [ch\\_options](#)

### 12.75.1 Detailed Description

User modifiable agency [options](#).

Definition at line 248 of file libmc.h.

### 12.75.2 Field Documentation

#### 12.75.2.1 ChOptions\_t\* MCAgencyOptions\_s::ch\_options

Definition at line 261 of file libmc.h.

Referenced by MC\_Initialize().

#### 12.75.2.2 int MCAgencyOptions\_s::default\_agent\_status

Default agent status

Definition at line 250 of file libmc.h.

Referenced by MC\_Initialize(), and MC\_InitializeAgencyOptions().

#### 12.75.2.3 int MCAgencyOptions\_s::enable\_security

security enable flag

Definition at line 252 of file libmc.h.

#### 12.75.2.4 int MCAgencyOptions\_s::initInterps

Definition at line 260 of file libmc.h.

Referenced by MC\_Initialize(), and MC\_InitializeAgencyOptions().

**12.75.2.5 char\* MCAgencyOptions\_s::known\_host\_filename**

Definition at line 258 of file libmc.h.

Referenced by MC\_Initialize(), and MC\_InitializeAgencyOptions().

**12.75.2.6 int MCAgencyOptions\_s::modified**

unused

Definition at line 251 of file libmc.h.

Referenced by MC\_InitializeAgencyOptions().

**12.75.2.7 unsigned char MCAgencyOptions\_s::passphrase[32]**

security enable flag

Definition at line 253 of file libmc.h.

Referenced by MC\_Initialize(), and MC\_InitializeAgencyOptions().

**12.75.2.8 char\* MCAgencyOptions\_s::priv\_key\_filename**

Definition at line 259 of file libmc.h.

Referenced by MC\_Initialize(), and MC\_InitializeAgencyOptions().

**12.75.2.9 int MCAgencyOptions\_s::stack\_size[MC\_THREAD\_ALL]**

If the stack size is -1, use system def.

Definition at line 256 of file libmc.h.

Referenced by MC\_Initialize(), and MC\_InitializeAgencyOptions().

**12.75.2.10 int MCAgencyOptions\_s::threads**

Threads to start

Definition at line 249 of file libmc.h.

Referenced by MC\_Initialize(), MC\_InitializeAgencyOptions(), MC\_SetThreadOff(), MC\_SetThreadOn(), MC\_SetThreadsAllOff(), and MC\_SetThreadsAllOn().

The documentation for this struct was generated from the following file:

- /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/libmc.h

## 12.76 LibMC::MCAgency::MCAgencyOptions\_t Struct Reference

### Data Fields

- [int threads](#)
- [int default\\_agent\\_status](#)
- [int modified](#)
- [int enable\\_security](#)
- [int\[ \] stack\\_size](#)

### 12.76.1 Detailed Description

Definition at line 42 of file MCEExports.cs.

### 12.76.2 Field Documentation

#### 12.76.2.1 int LibMC::MCAgency::MCAgencyOptions\_t::default\_agent\_status

Default agent status

Definition at line 45 of file MCEExports.cs.

#### 12.76.2.2 int LibMC::MCAgency::MCAgencyOptions\_t::enable\_security

security enable flag

Definition at line 47 of file MCEExports.cs.

#### 12.76.2.3 int LibMC::MCAgency::MCAgencyOptions\_t::modified

unused

Definition at line 46 of file MCEExports.cs.

#### 12.76.2.4 int [ ] LibMC::MCAgency::MCAgencyOptions\_t::stack\_size

If the stack size is -1, use system def.

Definition at line 50 of file MCEExports.cs.

#### 12.76.2.5 int LibMC::MCAgency::MCAgencyOptions\_t::threads

Threads to start

Definition at line 44 of file MCEExports.cs.

The documentation for this struct was generated from the following file:

- </home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/win32/LibMC.net/LibMC/MCEExports.cs>

## 12.77 LibMC::MCAgent Class Reference

Wrapper class for MCAgent\_t structure.

### Public Types

- enum [MC\\_AgentType\\_e](#) { [MC\\_NONE](#) = -1, [MC\\_REMOTE\\_AGENT](#) = 0, [MC\\_LOCAL\\_AGENT](#), [MC\\_RETURN\\_AGENT](#) }  
*Enum for describing the type of an agent.*
- enum [MC\\_AgentStatus\\_e](#) {  
[MC\\_NO\\_STATUS](#) = -1, [MC\\_WAIT\\_CH](#) = 0, [MC\\_WAIT\\_MESSGSEND](#), [MC\\_AGENT\\_ACTIVE](#),  
[MC\\_AGENT\\_NEUTRAL](#), [MC\\_AGENT\\_SUSPENDED](#), [MC\\_WAIT\\_FINISHED](#) }  
*Enum for describing the status of an agent.*

### Public Member Functions

- [MCAgent](#) ()  
*Default constructor.*
- override string [ToString](#) ()  
*Display the agent's fields.*
- int [DeleteAgent](#) ()  
*Deletes an agent.*
- String [GetAgentXMLString](#) ()  
*Gets the agent's XML string.*
- int [PrintAgentCode](#) ()  
*Gets the agent's C code string.*
- String [RetrieveAgentCode](#) ()  
*Gets the agent's C code string.*
- int [TerminateAgent](#) ()  
*Terminates an agent.*
- int [AclPost](#) ([MCAclMessage](#) message)  
*Posts an ACL message to the agent.*
- [MCAclMessage](#) [AclRetrieve](#) ()  
*Retrieve an ACL message from the agent.*
- [MCAclMessage](#) [AclWaitRetrieve](#) ()  
*Wait for and retrieve an ACL message from the agent.*

- [int CallAgentFunc](#) (String funcName, IntPtr retval, IntPtr varg)  
*Calls a function in an agent script.*
- [int CallAgentFunc](#) (String funcName, ref object retval, ref object varg)  
*Calls a function in an agent script.*
- IntPtr [GetAgentExecEngine](#) ()  
*Gets an agent's Ch interpreter.*
- [int GetAgentReturnData](#) (int task\_num, IntPtr data, IntPtr dim, IntPtr extent)  
*Calls a function in an agent script.*

## Static Public Member Functions

- static implicit [operator IntPtr](#) (MCAgent agent)
- static implicit [operator MCAgent](#) (IntPtr ip)

## Properties

- internal IntPtr [Agent](#) [get, set]
- [int AgentID](#) [get]  
*Gets the agent's ID number.*
- String [AgentName](#) [get]  
*Gets the agent's name.*
- [int AgentNumTasks](#) [get]  
*Gets the agent's number of tasks.*
- [MC\\_AgentStatus\\_e AgentStatus](#) [get, set]  
*Gets or sets the agent's status.*
- [MC\\_AgentType\\_e AgentType](#) [get]  
*Gets the agent's type.*
- bool [Valid](#) [get]  
*Checks whether the agent is valid.*

## Private Member Functions

- internal [MCAgent](#) (IntPtr ip)
- void [GetAgentFields](#) ()

## Private Attributes

- IntPtr `agent_p`
- String `name` = ""
- int `id` = -1
- int `numTasks` = -1
- `MC_AgentStatus_e status` = `MC_AgentStatus_e.MC_NO_STATUS`
- `MC_AgentType_e type` = `MC_AgentType_e.MC_NONE`

### 12.77.1 Detailed Description

Wrapper class for `MCAgent_t` structure. This class provides an interface to the Mobile-C agent structure. Member functions for the class are generally overloaded versions of the respective functions in the Mobile-C library. The class maintains a pointer to a Mobile-C agent in unmanaged memory. The pointer is not accessible by the user.

Definition at line 61 of file `MCAgent.cs`.

### 12.77.2 Member Enumeration Documentation

#### 12.77.2.1 enum `LibMC::MCAgent::MC_AgentStatus_e`

Enum for describing the status of an agent.

##### Note:

This enum is pulled directly from the Mobile-C library.

##### Enumerator:

**`MC_NO_STATUS`** Default value for uninitialized agent  
**`MC_WAIT_CH`** Waiting to be started  
**`MC_WAIT_MESSGSEND`** Finished, waiting to migrate  
**`MC_AGENT_ACTIVE`** Running  
**`MC_AGENT_NEUTRAL`** Not running, but do not flush  
**`MC_AGENT_SUSPENDED`** Unused  
**`MC_WAIT_FINISHED`** Finished, waiting to be flushed

Definition at line 88 of file `MCAgent.cs`.

#### 12.77.2.2 enum `LibMC::MCAgent::MC_AgentType_e`

Enum for describing the type of an agent.

##### Note:

This enum is pulled directly from the Mobile-C library.

##### Enumerator:

**`MC_NONE`** Default value to describe unininitialized agent.



**MC\_REMOTE\_AGENT** A remote agent.

**MC\_LOCAL\_AGENT** A local agent.

**MC\_RETURN\_AGENT** A returning agent.

Definition at line 75 of file MCAgent.cs.

### 12.77.3 Constructor & Destructor Documentation

#### 12.77.3.1 LibMC::MCAgent::MCAgent () [inline]

Default constructor. Creates an empty agent.

Definition at line 104 of file MCAgent.cs.

Referenced by operator MCAgent().

#### 12.77.3.2 internal LibMC::MCAgent::MCAgent (IntPtr ip) [inline, private]

Definition at line 110 of file MCAgent.cs.

References Agent.

### 12.77.4 Member Function Documentation

#### 12.77.4.1 int LibMC::MCAgent::AclPost (MCAclMessage message) [inline]

Posts an ACL message to the agent. Delivers an ACL message to the agent.

##### Parameters:

*message* The ACL message object to deliver.

##### Returns:

The return value of the underlying MC\_AclPost function call.

##### Note:

The message must be a valid message or this function call will fail.

Definition at line 397 of file MCAgent.cs.

References LibMC::MCAclMessage::AclMsg, and Agent.

#### 12.77.4.2 MCAclMessage LibMC::MCAgent::AclRetrieve () [inline]

Retrieve an ACL message from the agent. Retrieves an ACL message from the agent if one is available.

##### Returns:

The ACL message or a blank ACL message if one was not available.

**Note:**

The message must be a valid message or this function call will fail.

Definition at line 413 of file MCAgent.cs.

References Agent.

**12.77.4.3 MCAclMessage LibMC::MCAgent::AclWaitRetrieve () [inline]**

Wait for and retrieve an ACL message from the agent. Retrieves an ACL message from the agent when one becomes available.

**Returns:**

The ACL message or a blank ACL message if the call fails.

**Note:**

This function call blocks.

Definition at line 432 of file MCAgent.cs.

References Agent.

**12.77.4.4 int LibMC::MCAgent::CallAgentFunc (String *funcName*, ref object *retval*, ref object *varg*) [inline]**

Calls a function in an agent script. Calls a function in an agent's script file. This function requires boxing of parameters, but no marshaling.

**Parameters:**

*funcName* The name of the function to call  
*retval* A boxed object to hold the return value  
*varg* The boxed argument to the agent function

**Returns:**

The return value of the underlying MC\_CallAgentFunc function call.

**Note:**

This function handles marshaling of the argument and return value. The memory provided to the agent function for both *retval* and *varg* is not preserved after this function call! If the memory is to be kept by the agent, use the manually marshaled version of this function. Also note that even though structures can be marshaled automatically, in this function, the type of the structure is unknown and therefore it must be handled manually, even though the marshaling is transparent to the user.

**See also:**

Overloaded [CallAgentFunc](#), LibMCConsole example

Definition at line 488 of file MCAgent.cs.

References CallAgentFunc().

**12.77.4.5 int LibMC::MCAgent::CallAgentFunc (String *funcName*, IntPtr *retval*, IntPtr *varg*) [inline]**

Calls a function in an agent script. Calls a function in an agent's script file. This function requires manual marshaling by the user.

**Parameters:**

*funcName* The name of the function to call  
*retval* A pointer to memory for the return value  
*varg* A pointer to the argument for the function

**Returns:**

The return value of the underlying MC\_CallAgentFunc function call.

**Note:**

BE VERY CAREFUL! You must marshal your arguments! If possible, use the other CallAgentFunc that handles marshaling automatically.

**See also:**

Overloaded [CallAgentFunc](#), LibMCConsole example

Definition at line 458 of file MCAgent.cs.

References Agent.

Referenced by CallAgentFunc().

**12.77.4.6 int LibMC::MCAgent::DeleteAgent () [inline]**

Deletes an agent. Deletes an agent from the agency.

**Returns:**

The return value of the underlying MC\_DeleteAgent function call.

Definition at line 321 of file MCAgent.cs.

References Agent.

**12.77.4.7 IntPtr LibMC::MCAgent::GetAgentExecEngine () [inline]**

Gets an agent's Ch interpreter. Gets a pointer to the agent's Ch interpreter. Will be improved shortly.

**Returns:**

A pointer to the Ch interpreter.

**Note:**

Nothing in the LibMC.NET library can make use of the Ch interpreter yet.

**Todo**

Wrap MC\_GetAgentExecEngine with an object for the void\* pointer return type (Ch interpreter).

Definition at line 533 of file MCAgent.cs.

References Agent.

**12.77.4.8 void LibMC::MCAgent::GetAgentFields () [inline, private]**

Definition at line 163 of file MCAgent.cs.

References Agent, name, numTasks, status, and type.

**12.77.4.9 int LibMC::MCAgent::GetAgentReturnData (int *task\_num*, IntPtr *data*, IntPtr *dim*, IntPtr *extent*) [inline]**

Calls a function in an agent script. Calls a function in an agent's script file. This function requires manual marshaling by the user.

**Parameters:**

*task\_num* Task number to get data from

*data* A pointer to memory for the data

*dim* A pointer to hold the dimensions of the data

*extent* A pointer to hold the dimensions of the data

**Returns:**

The return value of the underlying MC\_GetAgentReturnData function call.

**Note:**

This function does nothing but throw an exception right now.

**Todo**

Implement GetAgentReturnData

Definition at line 554 of file MCAgent.cs.

**12.77.4.10 String LibMC::MCAgent::GetAgentXMLString () [inline]**

Gets the agent's XML string. Returns the full XML string associated with the agent.

**Returns:**

The return value of the underlying MC\_GetAgentXMLString function call.

Definition at line 334 of file MCAgent.cs.

References Agent.

**12.77.4.11 static implicit LibMC::MCAgent::operator IntPtr (MCAgent *agent*) [inline, static]**

Definition at line 299 of file MCAgent.cs.

References Agent.

**12.77.4.12 static implicit LibMC::MCAgent::operator MCAgent (IntPtr *ip*) [inline, static]**

Definition at line 304 of file MCAgent.cs.

References MCAgent().

**12.77.4.13 int LibMC::MCAgent::PrintAgentCode () [inline]**

Gets the agent's C code string. Prints the C code associated with the agent to stdout.

**Returns:**

The return value of the underlying MC\_PrintAgentCode function call.

Definition at line 347 of file MCAgent.cs.

References Agent.

**12.77.4.14 String LibMC::MCAgent::RetrieveAgentCode () [inline]**

Gets the agent's C code string. Returns the C code associated with the agent.

**Returns:**

A string containing the agent's C code.

Definition at line 359 of file MCAgent.cs.

References Agent.

**12.77.4.15 int LibMC::MCAgent::TerminateAgent () [inline]**

Terminates an agent. Terminates an agent regardless of the agent's state.

**Returns:**

The return value of the underlying MC\_TerminateAgent function call.

Definition at line 372 of file MCAgent.cs.

References Agent.

**12.77.4.16 override string LibMC::MCAgent::ToString () [inline]**

Display the agent's fields. Formats and returns a string with all of the agent's properties.

**Returns:**

A string containing a formatted representation of the agent's properties.

**Note:**

The agency port and any other [options](#) must be set before calling this function.

Definition at line 126 of file MCAgent.cs.

References AgentID, AgentName, AgentNumTasks, AgentStatus, and AgentType.

## 12.77.5 Field Documentation

### 12.77.5.1 IntPtr LibMC::MCAgent::agent\_p [private]

Definition at line 63 of file MCAgent.cs.

### 12.77.5.2 int LibMC::MCAgent::id = -1 [private]

Definition at line 65 of file MCAgent.cs.

### 12.77.5.3 String LibMC::MCAgent::name = "" [private]

Definition at line 64 of file MCAgent.cs.

Referenced by GetAgentFields().

### 12.77.5.4 int LibMC::MCAgent::numTasks = -1 [private]

Definition at line 66 of file MCAgent.cs.

Referenced by GetAgentFields().

### 12.77.5.5 MC\_AgentStatus\_e LibMC::MCAgent::status = MC\_AgentStatus\_e.MC\_NO\_STATUS [private]

Definition at line 67 of file MCAgent.cs.

Referenced by GetAgentFields().

### 12.77.5.6 MC\_AgentType\_e LibMC::MCAgent::type = MC\_AgentType\_e.MC\_NONE [private]

Definition at line 68 of file MCAgent.cs.

Referenced by GetAgentFields().

## 12.77.6 Property Documentation

### 12.77.6.1 internal IntPtr LibMC::MCAgent::Agent [get, set, private]

Definition at line 143 of file MCAgent.cs.

Referenced by AclPost(), AclRetrieve(), AclWaitRetrieve(), LibMC::MCAgency::AddAgent(), CallAgentFunc(), DeleteAgent(), GetAgentExecEngine(), GetAgentFields(), GetAgentXMLString(), MCAgent(), operator IntPtr(), PrintAgentCode(), LibMC::MCAgency::RegisterService(), RetrieveAgentCode(), and TerminateAgent().

#### 12.77.6.2 int LibMC::MCAgent::AgentID [get]

Gets the agent's ID number. Gets the agent's ID number as assigned by Mobile-C if the agent is a valid agent.

##### Returns:

The agent's ID number or -1 for an empty agent.

Definition at line 181 of file MCAgent.cs.

Referenced by ToString().

#### 12.77.6.3 String LibMC::MCAgent::AgentName [get]

Gets the agent's name. Gets the agent's name as assigned by Mobile-C or the agent script if the agent is a valid agent.

##### Returns:

The agent's name or an empty string for an empty agent.

Definition at line 201 of file MCAgent.cs.

Referenced by ToString().

#### 12.77.6.4 int LibMC::MCAgent::AgentNumTasks [get]

Gets the agent's number of tasks. Gets the agent's ID number of tasks if the agent is a valid agent.

##### Returns:

The agent's ID number of tasks or -1 for an empty agent.

Definition at line 220 of file MCAgent.cs.

Referenced by ToString().

#### 12.77.6.5 MC\_AgentStatus\_e LibMC::MCAgent::AgentStatus [get, set]

Gets or sets the agent's status. Gets or sets the agent's status. When setting the status, the status is double-checked after setting it and may not be set depending on the state of the agent and the agency.

##### Returns:

The agent's status or MC\_NO\_STATUS for an empty agent.

Definition at line 242 of file MCAgent.cs.

Referenced by ToString().

**12.77.6.6 MC\_AgentType\_e LibMC::MCAgent::AgentType [get]**

Gets the agent's type. Gets the agent's type.

**Returns:**

The agent's type or MC\_NONE for an empty agent.

Definition at line 264 of file MCAgent.cs.

Referenced by ToString().

**12.77.6.7 bool LibMC::MCAgent::Valid [get]**

Checks whether the agent is valid. Checks the internal agent pointer to see if it is non-zero.

**Returns:**

True if the pointer is valid, false otherwise

Definition at line 285 of file MCAgent.cs.

The documentation for this class was generated from the following file:

- </home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/win32/LibMC.net/LibMC/MCAgent.cs>



## 12.78 md2\_context Struct Reference

MD2 context structure.

```
#include <md2.h>
```

### Data Fields

- unsigned char [cksum](#) [16]
- unsigned char [state](#) [48]
- unsigned char [buffer](#) [16]
- unsigned char [ipad](#) [64]
- unsigned char [opad](#) [64]
- [int](#) [left](#)

### 12.78.1 Detailed Description

MD2 context structure.

Definition at line 10 of file md2.h.

### 12.78.2 Field Documentation

#### 12.78.2.1 unsigned char md2\_context::buffer[16]

data block being processed

Definition at line 14 of file md2.h.

#### 12.78.2.2 unsigned char md2\_context::cksum[16]

checksum of the data block

Definition at line 12 of file md2.h.

#### 12.78.2.3 unsigned char md2\_context::ipad[64]

HMAC: inner padding

Definition at line 16 of file md2.h.

#### 12.78.2.4 int md2\_context::left

amount of data in buffer

Definition at line 18 of file md2.h.

#### 12.78.2.5 unsigned char md2\_context::opad[64]

HMAC: outer padding

Definition at line 17 of file md2.h.

**12.78.2.6 unsigned char md2\_context::state[48]**

intermediate digest state

Definition at line 13 of file md2.h.

The documentation for this struct was generated from the following file:

- </home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/md2.h>

## 12.79 md4\_context Struct Reference

MD4 context structure.

```
#include <md4.h>
```

### Data Fields

- unsigned long [total](#) [2]
- unsigned long [state](#) [4]
- unsigned char [buffer](#) [64]
- unsigned char [ipad](#) [64]
- unsigned char [opad](#) [64]

### 12.79.1 Detailed Description

MD4 context structure.

Definition at line 10 of file md4.h.

### 12.79.2 Field Documentation

#### 12.79.2.1 unsigned char md4\_context::buffer[64]

data block being processed

Definition at line 14 of file md4.h.

#### 12.79.2.2 unsigned char md4\_context::ipad[64]

HMAC: inner padding

Definition at line 16 of file md4.h.

#### 12.79.2.3 unsigned char md4\_context::opad[64]

HMAC: outer padding

Definition at line 17 of file md4.h.

#### 12.79.2.4 unsigned long md4\_context::state[4]

intermediate digest state

Definition at line 13 of file md4.h.

#### 12.79.2.5 unsigned long md4\_context::total[2]

number of bytes processed

Definition at line 12 of file md4.h.

The documentation for this struct was generated from the following file:

- </home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/md4.h>

## 12.80 md5\_context Struct Reference

MD5 context structure.

```
#include <md5.h>
```

### Data Fields

- unsigned long [total](#) [2]
- unsigned long [state](#) [4]
- unsigned char [buffer](#) [64]
- unsigned char [ipad](#) [64]
- unsigned char [opad](#) [64]

### 12.80.1 Detailed Description

MD5 context structure.

Definition at line 10 of file md5.h.

### 12.80.2 Field Documentation

#### 12.80.2.1 unsigned char md5\_context::buffer[64]

data block being processed

Definition at line 14 of file md5.h.

Referenced by md5\_update().

#### 12.80.2.2 unsigned char md5\_context::ipad[64]

HMAC: inner padding

Definition at line 16 of file md5.h.

Referenced by md5\_hmac\_starts().

#### 12.80.2.3 unsigned char md5\_context::opad[64]

HMAC: outer padding

Definition at line 17 of file md5.h.

Referenced by md5\_hmac\_finish(), and md5\_hmac\_starts().

#### 12.80.2.4 unsigned long md5\_context::state[4]

intermediate digest state

Definition at line 13 of file md5.h.

Referenced by md5\_finish(), md5\_process(), md5\_starts(), and ssl\_calc\_finished().

**12.80.2.5 unsigned long md5\_context::total[2]**

number of bytes processed

Definition at line 12 of file md5.h.

Referenced by md5\_finish(), md5\_starts(), and md5\_update().

The documentation for this struct was generated from the following file:

- </home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/md5.h>

## 12.81 message\_s Struct Reference

```
#include <message.h>
```

### Data Fields

- struct sockaddr\_in \* [addr](#)
- [int connect\\_id](#)
- [int message\\_id](#)
- [int isHTTP](#)
- enum [message\\_type\\_e](#) [message\\_type](#)
- enum [http\\_performative\\_e](#) [http\\_type](#)
- [mxml\\_node\\_t](#) \* [xml\\_root](#)
- [mxml\\_node\\_t](#) \* [xml\\_payload](#)
- char \* [message\\_body](#)
- char \* [update\\_name](#)
- [int update\\_num](#)
- char \* [from\\_address](#)
- char \* [to\\_address](#)
- char \* [target](#)
- [int agent\\_xml\\_flag](#)
- char \* [sending\\_agent\\_name](#)

### 12.81.1 Detailed Description

Definition at line 77 of file message.h.

### 12.81.2 Field Documentation

#### 12.81.2.1 struct sockaddr\_in\* message\_s::addr [read]

Definition at line 79 of file message.h.

Referenced by [message\\_Destroy\(\)](#), [message\\_InitializeFromAgent\(\)](#), [message\\_InitializeFromConnection\(\)](#), [message\\_InitializeFromString\(\)](#), [message\\_New\(\)](#), and [message\\_xml\\_parse\\_\\_message\(\)](#).

#### 12.81.2.2 int message\_s::agent\_xml\_flag

Definition at line 111 of file message.h.

Referenced by [agent\\_Initialize\(\)](#), [message\\_Destroy\(\)](#), [message\\_InitializeFromAgent\(\)](#), and [message\\_New\(\)](#).

#### 12.81.2.3 int message\_s::connect\_id

Definition at line 82 of file message.h.

Referenced by [message\\_InitializeFromConnection\(\)](#), [message\\_InitializeFromString\(\)](#), and [message\\_New\(\)](#).

#### 12.81.2.4 `char* message_s::from_address`

Definition at line 101 of file message.h.

Referenced by `AP_QUEUE_SEARCH_TEMPLATE()`, `message_Destroy()`, `message_InitializeFromAgent()`, `message_InitializeFromConnection()`, `message_InitializeFromString()`, `message_New()`, and `message_xml_parse__message()`.

#### 12.81.2.5 `enum http_performative_e message_s::http_type`

Definition at line 90 of file message.h.

Referenced by `message_New()`.

#### 12.81.2.6 `int message_s::isHTTP`

Definition at line 86 of file message.h.

Referenced by `message_New()`, `mtp_http_ComposeMessage()`, and `mtp_http_CreateMessage()`.

#### 12.81.2.7 `char* message_s::message_body`

Definition at line 96 of file message.h.

Referenced by `acc_connection_Thread()`, `message_Destroy()`, `message_InitializeFromAgent()`, `message_InitializeFromConnection()`, `message_InitializeFromString()`, `message_New()`, `message_send_Thread()`, `mtp_http_ComposeMessage()`, and `mtp_http_CreateMessage()`.

#### 12.81.2.8 `int message_s::message_id`

Definition at line 83 of file message.h.

Referenced by `AP_QUEUE_SEARCH_TEMPLATE()`, `message_InitializeFromAgent()`, `message_InitializeFromConnection()`, `message_InitializeFromString()`, and `message_New()`.

#### 12.81.2.9 `enum message_type_e message_s::message_type`

Definition at line 89 of file message.h.

Referenced by `acc_connection_Thread()`, `acc_MessageHandlerThread()`, `agent_Initialize()`, `MC_AclSend()`, `message_InitializeFromAgent()`, `message_InitializeFromString()`, `message_New()`, and `message_xml_parse__message()`.

#### 12.81.2.10 `char* message_s::sending_agent_name`

Definition at line 112 of file message.h.

Referenced by `message_InitializeFromAgent()`, and `message_send_Thread()`.

#### 12.81.2.11 `char* message_s::target`

Definition at line 104 of file message.h.



Referenced by MC\_AclSend(), message\_Destroy(), message\_InitializeFromAgent(), message\_InitializeFromConnection(), message\_InitializeFromString(), message\_New(), and mtp\_http\_ComposeMessage().

#### 12.81.2.12 char\* message\_s::to\_address

Definition at line 102 of file message.h.

Referenced by acc\_MessageHandlerThread(), AP\_QUEUE\_SEARCH\_TEMPLATE(), MC\_LoadAgentFromFile(), message\_Destroy(), message\_InitializeFromAgent(), message\_InitializeFromConnection(), message\_InitializeFromString(), message\_New(), message\_queue\_SendOutgoing(), message\_send\_Thread(), mtp\_http\_ComposeMessage(), and mtp\_http\_CreateMessage().

#### 12.81.2.13 char\* message\_s::update\_name

Definition at line 98 of file message.h.

Referenced by message\_Destroy(), message\_InitializeFromAgent(), message\_InitializeFromString(), and message\_New().

#### 12.81.2.14 int message\_s::update\_num

Definition at line 99 of file message.h.

Referenced by message\_New().

#### 12.81.2.15 mxml\_node\_t\* message\_s::xml\_payload

Definition at line 93 of file message.h.

Referenced by agent\_Initialize(), MC\_LoadAgentFromFile(), message\_New(), and message\_xml\_parse\_message().

#### 12.81.2.16 mxml\_node\_t\* message\_s::xml\_root

Definition at line 92 of file message.h.

Referenced by acc\_connection\_Thread(), agent\_Initialize(), MC\_LoadAgentFromFile(), message\_Destroy(), message\_InitializeFromAgent(), message\_InitializeFromConnection(), message\_InitializeFromString(), message\_New(), and message\_xml\_parse().

The documentation for this struct was generated from the following file:

- /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/[message.h](#)

## 12.82 message\_send\_arg\_s Struct Reference

```
#include <message.h>
```

### Data Fields

- struct [mc\\_platform\\_s](#) \* [mc\\_platform](#)
- [message\\_p](#) [message](#)
- char \* [privatekey](#)

### 12.82.1 Detailed Description

Definition at line 116 of file [message.h](#).

### 12.82.2 Field Documentation

#### 12.82.2.1 struct [mc\\_platform\\_s](#)\* [message\\_send\\_arg\\_s::mc\\_platform](#) [read]

Definition at line 118 of file [message.h](#).

Referenced by [message\\_Send\(\)](#).

#### 12.82.2.2 [message\\_p](#) [message\\_send\\_arg\\_s::message](#)

Definition at line 119 of file [message.h](#).

Referenced by [message\\_Send\(\)](#).

#### 12.82.2.3 char\* [message\\_send\\_arg\\_s::privatekey](#)

Definition at line 120 of file [message.h](#).

Referenced by [message\\_Send\(\)](#).

The documentation for this struct was generated from the following file:

- [/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/message.h](#)

## 12.83 mpi Struct Reference

MPI structure.

```
#include <bignum.h>
```

### Data Fields

- `int s`
- `int n`
- `t_int * p`

### 12.83.1 Detailed Description

MPI structure.

Definition at line 48 of file `bignum.h`.

### 12.83.2 Field Documentation

#### 12.83.2.1 `int mpi::n`

total # of limbs

Definition at line 51 of file `bignum.h`.

Referenced by `debug_print_mpi()`, `mpi_add_abs()`, `mpi_add_int()`, `mpi_cmp_abs()`, `mpi_cmp_int()`, `mpi_cmp_mpi()`, `mpi_copy()`, `mpi_div_int()`, `mpi_div_mpi()`, `mpi_exp_mod()`, `mpi_free()`, `mpi_grow()`, `mpi_init()`, `mpi_is_prime()`, `mpi_lsb()`, `mpi_lset()`, `mpi_mod_int()`, `mpi_montmul()`, `mpi_montred()`, `mpi_msb()`, `mpi_mul_int()`, `mpi_mul_mpi()`, `mpi_shift_l()`, `mpi_shift_r()`, `mpi_sub_abs()`, `mpi_sub_int()`, `mpi_write_string()`, and `x509parse_cert_info()`.

#### 12.83.2.2 `t_int* mpi::p`

pointer to limbs

Definition at line 52 of file `bignum.h`.

Referenced by `debug_print_mpi()`, `dhm_make_params()`, `dhm_make_public()`, `mpi_add_abs()`, `mpi_add_int()`, `mpi_cmp_abs()`, `mpi_cmp_int()`, `mpi_cmp_mpi()`, `mpi_copy()`, `mpi_div_int()`, `mpi_div_mpi()`, `mpi_exp_mod()`, `mpi_free()`, `mpi_gcd()`, `mpi_gen_prime()`, `mpi_grow()`, `mpi_init()`, `mpi_inv_mod()`, `mpi_is_prime()`, `mpi_lsb()`, `mpi_lset()`, `mpi_mod_int()`, `mpi_montg_init()`, `mpi_montmul()`, `mpi_montred()`, `mpi_msb()`, `mpi_mul_int()`, `mpi_mul_mpi()`, `mpi_read_binary()`, `mpi_read_string()`, `mpi_shift_l()`, `mpi_shift_r()`, `mpi_sub_abs()`, `mpi_sub_int()`, `mpi_write_binary()`, `mpi_write_string()`, and `rsa_check_pubkey()`.

#### 12.83.2.3 `int mpi::s`

integer sign

Definition at line 50 of file `bignum.h`.

Referenced by `mpi_add_int()`, `mpi_add_mpi()`, `mpi_cmp_int()`, `mpi_cmp_mpi()`, `mpi_copy()`, `mpi_div_int()`, `mpi_div_mpi()`, `mpi_free()`, `mpi_gcd()`, `mpi_init()`, `mpi_is_prime()`, `mpi_lset()`, `mpi_montred()`,

`mpi_mul_int()`, `mpi_mul_mpi()`, `mpi_read_string()`, `mpi_sub_int()`, `mpi_sub_mpi()`, and `mpi_write_string()`.

The documentation for this struct was generated from the following file:

- </home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/bignum.h>

## 12.84 mtp\_http\_content\_s Struct Reference

```
#include <mtp_http.h>
```

### Data Fields

- char \* [content\\_type](#)
- void \* [data](#)

### 12.84.1 Detailed Description

Definition at line 106 of file mtp\_http.h.

### 12.84.2 Field Documentation

#### 12.84.2.1 char\* mtp\_http\_content\_s::content\_type

Definition at line 108 of file mtp\_http.h.

Referenced by MC\_AclSend(), mtp\_http\_CreateMessage(), mtp\_http\_Destroy(), and mtp\_http\_Parse().

#### 12.84.2.2 void\* mtp\_http\_content\_s::data

Definition at line 109 of file mtp\_http.h.

Referenced by acc\_connection\_Thread(), MC\_AclSend(), mtp\_http\_CreateMessage(), mtp\_http\_Destroy(), and mtp\_http\_Parse().

The documentation for this struct was generated from the following file:

- /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/[mtp\\_http.h](#)

## 12.85 mtp\_http\_s Struct Reference

```
#include <mtp_http.h>
```

### Data Fields

- enum [http\\_status\\_code\\_e](#) `http_status_code`
- enum [http\\_performative\\_e](#) `http_performative`
- char \* [http\\_version](#)
- char \* [host](#)
- char \* [return\\_code](#)
- char \* [target](#)
- char \* [date](#)
- char \* [server](#)
- char \* [accept\\_ranges](#)
- char \* [content\\_length](#)
- char \* [connection](#)
- char \* [content\\_type](#)
- char \* [user\\_agent](#)
- char \* [cache\\_control](#)
- char \* [mime\\_version](#)
- int [response\\_code](#)
- char \* [response\\_string](#)
- int [message\\_parts](#)
- char \* [boundary](#)
- struct [mtp\\_http\\_content\\_s](#) \* `content`
- int [header\\_length](#)

### 12.85.1 Detailed Description

Definition at line 112 of file `mtp_http.h`.

### 12.85.2 Field Documentation

#### 12.85.2.1 char\* mtp\_http\_s::accept\_ranges

Definition at line 123 of file `mtp_http.h`.

Referenced by `mtp_http_Destroy()`.

#### 12.85.2.2 char\* mtp\_http\_s::boundary

Definition at line 139 of file `mtp_http.h`.

Referenced by `mtp_http_Destroy()`, and `mtp_http_Parse()`.

#### 12.85.2.3 char\* mtp\_http\_s::cache\_control

Definition at line 129 of file `mtp_http.h`.

**12.85.2.4 char\* mtp\_http\_s::connection**

Definition at line 125 of file mtp\_http.h.

Referenced by mtp\_http\_Destroy().

**12.85.2.5 struct mtp\_http\_content\_s\* mtp\_http\_s::content [read]**

Definition at line 140 of file mtp\_http.h.

Referenced by acc\_connection\_Thread(), MC\_AclSend(), mtp\_http\_CreateMessage(), mtp\_http\_Destroy(), mtp\_http\_New(), and mtp\_http\_Parse().

**12.85.2.6 char\* mtp\_http\_s::content\_length**

Definition at line 124 of file mtp\_http.h.

Referenced by mtp\_http\_Destroy(), mtp\_http\_InitializeFromConnection(), and mtp\_http\_Parse().

**12.85.2.7 char\* mtp\_http\_s::content\_type**

Definition at line 126 of file mtp\_http.h.

Referenced by mtp\_http\_Destroy(), and mtp\_http\_Parse().

**12.85.2.8 char\* mtp\_http\_s::date**

Definition at line 121 of file mtp\_http.h.

Referenced by mtp\_http\_Destroy().

**12.85.2.9 int mtp\_http\_s::header\_length**

Definition at line 143 of file mtp\_http.h.

Referenced by mtp\_http\_InitializeFromConnection(), and mtp\_http\_ParseHeader().

**12.85.2.10 char\* mtp\_http\_s::host**

Definition at line 117 of file mtp\_http.h.

Referenced by MC\_AclSend(), mtp\_http\_CreateMessage(), and mtp\_http\_Destroy().

**12.85.2.11 enum http\_performative\_e mtp\_http\_s::http\_performative**

Definition at line 115 of file mtp\_http.h.

Referenced by acc\_connection\_Thread(), http\_ParseRequest(), mtp\_http\_InitializeFromConnection(), and mtp\_http\_Parse().

**12.85.2.12 enum http\_status\_code\_e mtp\_http\_s::http\_status\_code**

Definition at line 114 of file mtp\_http.h.

**12.85.2.13 char\* mtp\_http\_s::http\_version**

Definition at line 116 of file mtp\_http.h.

Referenced by mtp\_http\_Destroy().

**12.85.2.14 int mtp\_http\_s::message\_parts**

Definition at line 138 of file mtp\_http.h.

Referenced by acc\_connection\_Thread(), MC\_AclSend(), mtp\_http\_CreateMessage(), mtp\_http\_Destroy(), and mtp\_http\_Parse().

**12.85.2.15 char\* mtp\_http\_s::mime\_version**

Definition at line 130 of file mtp\_http.h.

**12.85.2.16 int mtp\_http\_s::response\_code**

Definition at line 133 of file mtp\_http.h.

Referenced by http\_ParseRequest().

**12.85.2.17 char\* mtp\_http\_s::response\_string**

Definition at line 134 of file mtp\_http.h.

Referenced by http\_ParseRequest(), and mtp\_http\_Destroy().

**12.85.2.18 char\* mtp\_http\_s::return\_code**

Definition at line 118 of file mtp\_http.h.

Referenced by mtp\_http\_Destroy().

**12.85.2.19 char\* mtp\_http\_s::server**

Definition at line 122 of file mtp\_http.h.

Referenced by mtp\_http\_Destroy().

**12.85.2.20 char\* mtp\_http\_s::target**

Definition at line 119 of file mtp\_http.h.

Referenced by acc\_connection\_Thread(), http\_ParseRequest(), MC\_AclSend(), mtp\_http\_CreateMessage(), and mtp\_http\_Destroy().



**12.85.2.21 char\* mtp\_http\_s::user\_agent**

Definition at line 127 of file mtp\_http.h.

Referenced by mtp\_http\_Destroy().

The documentation for this struct was generated from the following file:

- /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/[mtp\\_http.h](#)

## 12.86 mxml\_attr\_s Struct Reference

```
#include <mxml.h>
```

### Data Fields

- char \* [name](#)
- char \* [value](#)

### 12.86.1 Detailed Description

Definition at line 86 of file mxml.h.

### 12.86.2 Field Documentation

#### 12.86.2.1 char\* mxml\_attr\_s::name

Definition at line 88 of file mxml.h.

Referenced by [mxml\\_write\\_node\(\)](#), [mxmlDelete\(\)](#), [mxmlElementGetAttr\(\)](#), and [mxmlElementSetAttr\(\)](#).

#### 12.86.2.2 char\* mxml\_attr\_s::value

Definition at line 89 of file mxml.h.

Referenced by [mxml\\_write\\_node\(\)](#), [mxmlDelete\(\)](#), [mxmlElementGetAttr\(\)](#), and [mxmlElementSetAttr\(\)](#).

The documentation for this struct was generated from the following file:

- [/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mxml-2.2.2/mxml.h](#)

## 12.87 mxml\_custom\_s Struct Reference

```
#include <mxml.h>
```

### Data Fields

- void \* [data](#)
- void(\* [destroy](#) )(void \*)

### 12.87.1 Detailed Description

Definition at line 105 of file mxml.h.

### 12.87.2 Field Documentation

#### 12.87.2.1 void\* mxml\_custom\_s::data

Definition at line 107 of file mxml.h.

Referenced by mxmlDelete(), mxmlNewCustom(), and mxmlSetCustom().

#### 12.87.2.2 void(\* mxml\_custom\_s::destroy)(void \*)

Referenced by mxmlDelete(), mxmlNewCustom(), and mxmlSetCustom().

The documentation for this struct was generated from the following file:

- /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mxml-2.2.2/[mxml.h](#)

## 12.88 mxml\_fdbuf\_s Struct Reference

### Data Fields

- `int fd`
- `unsigned char * current`
- `unsigned char * end`
- `unsigned char buffer [8192]`

### 12.88.1 Detailed Description

Definition at line 83 of file mxml-file.c.

### 12.88.2 Field Documentation

#### 12.88.2.1 `unsigned char mxml_fdbuf_s::buffer[8192]`

Definition at line 86 of file mxml-file.c.

Referenced by `mxml_fd_read()`, `mxml_fd_write()`, `mxxmlLoadFd()`, and `mxxmlSaveFd()`.

#### 12.88.2.2 `unsigned char* mxml_fdbuf_s::current`

Definition at line 86 of file mxml-file.c.

Referenced by `mxml_fd_getc()`, `mxml_fd_putc()`, `mxml_fd_read()`, `mxml_fd_write()`, `mxxmlLoadFd()`, and `mxxmlSaveFd()`.

#### 12.88.2.3 `unsigned char * mxml_fdbuf_s::end`

Definition at line 86 of file mxml-file.c.

Referenced by `mxml_fd_getc()`, `mxml_fd_putc()`, `mxml_fd_read()`, `mxxmlLoadFd()`, and `mxxmlSaveFd()`.

#### 12.88.2.4 `int mxml_fdbuf_s::fd`

Definition at line 85 of file mxml-file.c.

Referenced by `mxml_fd_read()`, `mxml_fd_write()`, `mxxmlLoadFd()`, and `mxxmlSaveFd()`.

The documentation for this struct was generated from the following file:

- `/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mxml-2.2.2/mxml-file.c`

## 12.89 mxml\_index\_s Struct Reference

```
#include <mxml.h>
```

### Data Fields

- char \* [attr](#)
- int [num\\_nodes](#)
- int [alloc\\_nodes](#)
- int [cur\\_node](#)
- [mxml\\_node\\_t](#) \*\* [nodes](#)

### 12.89.1 Detailed Description

Definition at line 133 of file mxml.h.

### 12.89.2 Field Documentation

#### 12.89.2.1 int mxml\_index\_s::alloc\_nodes

Definition at line 137 of file mxml.h.

Referenced by [mxmlIndexDelete\(\)](#), and [mxmlIndexNew\(\)](#).

#### 12.89.2.2 char\* mxml\_index\_s::attr

Definition at line 135 of file mxml.h.

Referenced by [index\\_compare\(\)](#), [index\\_find\(\)](#), [mxmlIndexDelete\(\)](#), [mxmlIndexFind\(\)](#), and [mxmlIndexNew\(\)](#).

#### 12.89.2.3 int mxml\_index\_s::cur\_node

Definition at line 138 of file mxml.h.

Referenced by [mxmlIndexEnum\(\)](#), [mxmlIndexFind\(\)](#), and [mxmlIndexReset\(\)](#).

#### 12.89.2.4 mxml\_node\_t\*\* mxml\_index\_s::nodes

Definition at line 139 of file mxml.h.

Referenced by [index\\_sort\(\)](#), [mxmlIndexDelete\(\)](#), [mxmlIndexEnum\(\)](#), [mxmlIndexFind\(\)](#), [mxmlIndexNew\(\)](#), and [mxmlIndexReset\(\)](#).

#### 12.89.2.5 int mxml\_index\_s::num\_nodes

Definition at line 136 of file mxml.h.

Referenced by [main\(\)](#), [mxmlIndexEnum\(\)](#), [mxmlIndexFind\(\)](#), [mxmlIndexNew\(\)](#), and [mxmlIndexReset\(\)](#).

The documentation for this struct was generated from the following file:

- </home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mxml-2.2.2/mxml.h>

## 12.90 mxml\_node\_s Struct Reference

```
#include <mxml.h>
```

### Data Fields

- [mxml\\_type\\_t](#) type
- struct [mxml\\_node\\_s](#) \* next
- struct [mxml\\_node\\_s](#) \* prev
- struct [mxml\\_node\\_s](#) \* parent
- struct [mxml\\_node\\_s](#) \* child
- struct [mxml\\_node\\_s](#) \* last\_child
- [mxml\\_value\\_t](#) value

### 12.90.1 Detailed Description

Definition at line 122 of file mxml.h.

### 12.90.2 Field Documentation

#### 12.90.2.1 struct mxml\_node\_s\* mxml\_node\_s::child [read]

Definition at line 128 of file mxml.h.

Referenced by `add_variable()`, `agent_xml_parse__fill_row_data()`, `fipa_envelope_HandleAclRepresentation()`, `fipa_envelope_HandleComments()`, `fipa_envelope_HandleDate()`, `fipa_envelope_HandlePayloadEncoding()`, `fipa_envelope_HandlePayloadLength()`, `fipa_envelope_ParseAddresses()`, `fipa_envelope_ParseAgentIdentifier()`, `main()`, `mxml_write_node()`, `mxmlAdd()`, `mxmlDelete()`, `mxmlRemove()`, `mxmlWalkNext()`, `scan_file()`, `sort_node()`, `whitespace_cb()`, `write_documentation()`, `write_element()`, and `xml_get_text()`.

#### 12.90.2.2 struct mxml\_node\_s\* mxml\_node\_s::last\_child [read]

Definition at line 129 of file mxml.h.

Referenced by `add_variable()`, `main()`, `mxmlAdd()`, `mxmlRemove()`, `mxmlWalkPrev()`, and `scan_file()`.

#### 12.90.2.3 struct mxml\_node\_s\* mxml\_node\_s::next [read]

Definition at line 125 of file mxml.h.

Referenced by `add_variable()`, `main()`, `mxml_write_node()`, `mxmlAdd()`, `mxmlFindElement()`, `mxmlRemove()`, `mxmlWalkNext()`, `scan_file()`, `sort_node()`, `xml_get_next_element()`, and `xml_get_text()`.

#### 12.90.2.4 struct mxml\_node\_s\* mxml\_node\_s::parent [read]

Definition at line 127 of file mxml.h.

Referenced by `agent_xml_parse__data()`, `mxml_load_data()`, `mxmlAdd()`, `mxmlRemove()`, `mxmlWalkNext()`, `mxmlWalkPrev()`, `sort_node()`, `whitespace_cb()`, `ws_cb()`, and `xml_find_sibling()`.

### 12.90.2.5 struct mxml\_node\_s\* mxml\_node\_s::prev [read]

Definition at line 126 of file mxml.h.

Referenced by mxml\_write\_node(), mxmlAdd(), mxmlRemove(), and mxmlWalkPrev().

### 12.90.2.6 mxml\_type\_t mxml\_node\_s::type

Definition at line 124 of file mxml.h.

Referenced by agent\_xml\_parse\_\_fill\_row\_data(), fipa\_envelope\_HandleAclRepresentation(), fipa\_envelope\_HandleComments(), fipa\_envelope\_HandleDate(), fipa\_envelope\_HandlePayloadEncoding(), fipa\_envelope\_HandlePayloadLength(), fipa\_envelope\_ParseAddresses(), fipa\_envelope\_ParseAgentIdentifier(), main(), mxml\_new(), mxml\_write\_node(), mxmlDelete(), mxmlElementGetAttr(), mxmlElementSetAttr(), mxmlFindElement(), mxmlSetCustom(), mxmlSetElement(), mxmlSetInteger(), mxmlSetOpaque(), mxmlSetReal(), mxmlSetText(), mxmlSetTextf(), write\_element(), xml\_get\_cdata(), xml\_get\_element\_name(), xml\_get\_next\_element(), and xml\_get\_text().

### 12.90.2.7 mxml\_value\_t mxml\_node\_s::value

Definition at line 130 of file mxml.h.

Referenced by add\_variable(), agent\_xml\_parse\_\_fill\_row\_data(), fipa\_envelope\_HandleAclRepresentation(), fipa\_envelope\_HandleComments(), fipa\_envelope\_HandleDate(), fipa\_envelope\_HandlePayloadEncoding(), fipa\_envelope\_HandlePayloadLength(), fipa\_envelope\_ParseAddresses(), fipa\_envelope\_ParseAgentIdentifier(), index\_compare(), index\_find(), main(), mxml\_get\_entity(), mxml\_load\_data(), mxml\_parse\_element(), mxml\_write\_node(), mxmlDelete(), mxmlElementGetAttr(), mxmlElementSetAttr(), mxmlFindElement(), mxmlIndexNew(), mxmlNewCustom(), mxmlNewElement(), mxmlNewInteger(), mxmlNewOpaque(), mxmlNewReal(), mxmlNewText(), mxmlNewTextf(), mxmlSetCustom(), mxmlSetElement(), mxmlSetInteger(), mxmlSetOpaque(), mxmlSetReal(), mxmlSetText(), mxmlSetTextf(), scan\_file(), sort\_node(), type\_cb(), update\_comment(), whitespace\_cb(), write\_element(), ws\_cb(), xml\_get\_element\_name(), and xml\_get\_text().

The documentation for this struct was generated from the following file:

- /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mxml-2.2.2/[mxml.h](#)



## 12.91 mxml\_text\_s Struct Reference

```
#include <mxml.h>
```

### Data Fields

- [int whitespace](#)
- [char \\* string](#)

### 12.91.1 Detailed Description

Definition at line 99 of file mxml.h.

### 12.91.2 Field Documentation

#### 12.91.2.1 char\* mxml\_text\_s::string

Definition at line 102 of file mxml.h.

Referenced by `add_variable()`, `agent_xml_parse__fill_row_data()`, `fipa_envelope_HandleAclRepresentation()`, `fipa_envelope_HandleComments()`, `fipa_envelope_HandleDate()`, `fipa_envelope_HandlePayloadEncoding()`, `fipa_envelope_HandlePayloadLength()`, `fipa_envelope_ParseAddresses()`, `fipa_envelope_ParseAgentIdentifier()`, `main()`, `mxml_write_node()`, `mxmlDelete()`, `mxmlNewText()`, `mxmlNewTextf()`, `mxmlSetText()`, `mxmlSetTextf()`, `scan_file()`, `update_comment()`, `write_element()`, and `xml_get_text()`.

#### 12.91.2.2 int mxml\_text\_s::whitespace

Definition at line 101 of file mxml.h.

Referenced by `add_variable()`, `main()`, `mxml_write_node()`, `mxmlNewText()`, `mxmlNewTextf()`, `mxmlSetText()`, `mxmlSetTextf()`, `scan_file()`, and `write_element()`.

The documentation for this struct was generated from the following file:

- `/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mxml-2.2.2/mxml.h`

## 12.92 mxml\_value\_s Struct Reference

```
#include <mxml.h>
```

### Data Fields

- char \* [name](#)
- int [num\\_attrs](#)
- [mxml\\_attr\\_t](#) \* [attrs](#)

### 12.92.1 Detailed Description

Definition at line 92 of file mxml.h.

### 12.92.2 Field Documentation

#### 12.92.2.1 mxml\_attr\_t\* mxml\_value\_s::attrs

Definition at line 96 of file mxml.h.

Referenced by [mxml\\_write\\_node\(\)](#), [mxmlDelete\(\)](#), [mxmlElementGetAttr\(\)](#), and [mxmlElementSetAttr\(\)](#).

#### 12.92.2.2 char\* mxml\_value\_s::name

Definition at line 94 of file mxml.h.

Referenced by [index\\_compare\(\)](#), [index\\_find\(\)](#), [main\(\)](#), [mxml\\_get\\_entity\(\)](#), [mxml\\_load\\_data\(\)](#), [mxml\\_parse\\_element\(\)](#), [mxml\\_write\\_node\(\)](#), [mxmlDelete\(\)](#), [mxmlElementSetAttr\(\)](#), [mxmlFindElement\(\)](#), [mxmlIndexNew\(\)](#), [mxmlNewElement\(\)](#), [mxmlSetElement\(\)](#), [scan\\_file\(\)](#), [sort\\_node\(\)](#), [type\\_cb\(\)](#), [update\\_comment\(\)](#), [whitespace\\_cb\(\)](#), [ws\\_cb\(\)](#), [xml\\_get\\_element\\_name\(\)](#), and [xml\\_get\\_text\(\)](#).

#### 12.92.2.3 int mxml\_value\_s::num\_attrs

Definition at line 95 of file mxml.h.

Referenced by [mxml\\_write\\_node\(\)](#), [mxmlDelete\(\)](#), [mxmlElementGetAttr\(\)](#), and [mxmlElementSetAttr\(\)](#).

The documentation for this struct was generated from the following file:

- [/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mxml-2.2.2/mxml.h](#)

## 12.93 mxml\_value\_u Union Reference

```
#include <mxml.h>
```

### Data Fields

- [mxml\\_element\\_t](#) element
- [int](#) integer
- [char \\*](#) opaque
- [double](#) real
- [mxml\\_text\\_t](#) text
- [mxml\\_custom\\_t](#) custom

### 12.93.1 Detailed Description

Definition at line 112 of file mxml.h.

### 12.93.2 Field Documentation

#### 12.93.2.1 mxml\_custom\_t mxml\_value\_u::custom

Definition at line 119 of file mxml.h.

Referenced by [mxmlDelete\(\)](#), [mxmlNewCustom\(\)](#), and [mxmlSetCustom\(\)](#).

#### 12.93.2.2 mxml\_element\_t mxml\_value\_u::element

Definition at line 114 of file mxml.h.

Referenced by [index\\_compare\(\)](#), [index\\_find\(\)](#), [main\(\)](#), [mxml\\_get\\_entity\(\)](#), [mxml\\_load\\_data\(\)](#), [mxml\\_parse\\_element\(\)](#), [mxml\\_write\\_node\(\)](#), [mxmlDelete\(\)](#), [mxmlElementGetAttr\(\)](#), [mxmlElementSetAttr\(\)](#), [mxmlFindElement\(\)](#), [mxmlIndexNew\(\)](#), [mxmlNewElement\(\)](#), [mxmlSetElement\(\)](#), [scan\\_file\(\)](#), [sort\\_node\(\)](#), [type\\_cb\(\)](#), [update\\_comment\(\)](#), [whitespace\\_cb\(\)](#), [ws\\_cb\(\)](#), [xml\\_get\\_element\\_name\(\)](#), and [xml\\_get\\_text\(\)](#).

#### 12.93.2.3 int mxml\_value\_u::integer

Definition at line 115 of file mxml.h.

Referenced by [main\(\)](#), [mxml\\_write\\_node\(\)](#), [mxmlNewInteger\(\)](#), and [mxmlSetInteger\(\)](#).

#### 12.93.2.4 char\* mxml\_value\_u::opaque

Definition at line 116 of file mxml.h.

Referenced by [main\(\)](#), [mxml\\_write\\_node\(\)](#), [mxmlDelete\(\)](#), [mxmlNewOpaque\(\)](#), and [mxmlSetOpaque\(\)](#).

#### 12.93.2.5 double mxml\_value\_u::real

Definition at line 117 of file mxml.h.

Referenced by [main\(\)](#), [mxml\\_write\\_node\(\)](#), [mxmlNewReal\(\)](#), and [mxmlSetReal\(\)](#).

### 12.93.2.6 mxml\_text\_t mxml\_value\_u::text

Definition at line 118 of file mxml.h.

Referenced by add\_variable(), agent\_xml\_parse\_\_fill\_row\_data(), fipa\_envelope\_HandleAclRepresentation(), fipa\_envelope\_HandleComments(), fipa\_envelope\_HandleDate(), fipa\_envelope\_HandlePayloadEncoding(), fipa\_envelope\_HandlePayloadLength(), fipa\_envelope\_ParseAddresses(), fipa\_envelope\_ParseAgentIdentifier(), main(), mxml\_write\_node(), mxmlDelete(), mxmlNewText(), mxmlNewTextf(), mxmlSetText(), mxmlSetTextf(), scan\_file(), update\_comment(), write\_element(), and xml\_get\_text().

The documentation for this union was generated from the following file:

- /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mxml-2.2.2/[mxml.h](#)

## 12.94 options Struct Reference

### Data Fields

- `int opmode`
- `int iomode`
- `char * server_name`
- `int server_port`
- `int command`
- `int buffer_size`
- `int max_bytes`
- `int debug_level`
- `int conn_timeout`
- `int max_connections`
- `int session_reuse`
- `int session_lifetime`
- `int force_cipher` [2]

### 12.94.1 Detailed Description

Definition at line 75 of file `ssl_test.c`.

### 12.94.2 Field Documentation

#### 12.94.2.1 `int options::buffer_size`

Definition at line 82 of file `ssl_test.c`.

Referenced by `main()`, and `ssl_test()`.

#### 12.94.2.2 `int options::command`

Definition at line 81 of file `ssl_test.c`.

Referenced by `main()`, and `ssl_test()`.

#### 12.94.2.3 `int options::conn_timeout`

Definition at line 85 of file `ssl_test.c`.

Referenced by `main()`, and `ssl_test()`.

#### 12.94.2.4 `int options::debug_level`

Definition at line 84 of file `ssl_test.c`.

Referenced by `main()`.

**12.94.2.5 int options::force\_cipher[2]**

Definition at line 89 of file ssl\_test.c.

Referenced by main(), and ssl\_test().

**12.94.2.6 int options::iomode**

Definition at line 78 of file ssl\_test.c.

Referenced by main(), and ssl\_test().

**12.94.2.7 int options::max\_bytes**

Definition at line 83 of file ssl\_test.c.

Referenced by main(), and ssl\_test().

**12.94.2.8 int options::max\_connections**

Definition at line 86 of file ssl\_test.c.

Referenced by main().

**12.94.2.9 int options::opmode**

Definition at line 77 of file ssl\_test.c.

Referenced by main(), and ssl\_test().

**12.94.2.10 char\* options::server\_name**

Definition at line 79 of file ssl\_test.c.

Referenced by main(), and ssl\_test().

**12.94.2.11 int options::server\_port**

Definition at line 80 of file ssl\_test.c.

Referenced by main(), and ssl\_test().

**12.94.2.12 int options::session\_lifetime**

Definition at line 88 of file ssl\_test.c.

Referenced by main(), and ssl\_test().

**12.94.2.13 int options::session\_reuse**

Definition at line 87 of file ssl\_test.c.

Referenced by main(), and ssl\_test().

The documentation for this struct was generated from the following file:

- [/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/programs/test/ssl\\_test.c](#)

## 12.95 Program1::Program Class Reference

### Static Private Member Functions

- static void [Main](#) (string[ ] args)

### 12.95.1 Detailed Description

Definition at line 8 of file Program.cs.

### 12.95.2 Member Function Documentation

#### 12.95.2.1 static void Program1::Program::Main (string[ ] args) [inline, static, private]

Definition at line 10 of file Program.cs.

References [EmbeddedCh::ChInterp::AppendRunScript\(\)](#), [EmbeddedCh::ChInterp::AppendRunScriptFile\(\)](#), [EmbeddedCh::ChInterp::End\(\)](#), [EmbeddedCh::ChInterp::Initialize\(\)](#), and [EmbeddedCh::ChInterp::RunScript\(\)](#).

The documentation for this class was generated from the following file:

- [/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/win32/EmbeddedCh.Net/Program1/Program.cs](#)



## 12.96 rsa\_context Struct Reference

RSA context structure.

```
#include <rsa.h>
```

### Data Fields

- [int ver](#)
- [int len](#)
- [mpi N](#)
- [mpi E](#)
- [mpi D](#)
- [mpi P](#)
- [mpi Q](#)
- [mpi DP](#)
- [mpi DQ](#)
- [mpi QP](#)
- [mpi RN](#)
- [mpi RP](#)
- [mpi RQ](#)
- [int padding](#)
- [int hash\\_id](#)
- [int\(\\* f\\_rng\)\(void \\*\)](#)
- [void \\* p\\_rng](#)

### 12.96.1 Detailed Description

RSA context structure.

Definition at line 56 of file rsa.h.

### 12.96.2 Field Documentation

#### 12.96.2.1 mpi rsa\_context::D

private exponent

Definition at line 64 of file rsa.h.

Referenced by `generate_RSA_keys_ciphertext()`, `generate_RSA_keys_plaintext()`, `main()`, `rsa_check_privkey()`, `rsa_decryption()`, `rsa_free()`, `rsa_gen_key()`, `rsa_private()`, `rsa_self_test()`, and `x509parse_key()`.

#### 12.96.2.2 mpi rsa\_context::DP

$D \% (P - 1)$

Definition at line 67 of file rsa.h.

Referenced by `generate_RSA_keys_ciphertext()`, `generate_RSA_keys_plaintext()`, `main()`, `rsa_decryption()`, `rsa_free()`, `rsa_gen_key()`, `rsa_private()`, `rsa_self_test()`, and `x509parse_key()`.

### 12.96.2.3 `mpi rsa_context::DQ`

$D \% (Q - 1)$

Definition at line 68 of file `rsa.h`.

Referenced by `generate_RSA_keys_ciphertext()`, `generate_RSA_keys_plaintext()`, `main()`, `rsa_decryption()`, `rsa_free()`, `rsa_gen_key()`, `rsa_private()`, `rsa_self_test()`, and `x509parse_key()`.

### 12.96.2.4 `mpi rsa_context::E`

public exponent

Definition at line 62 of file `rsa.h`.

Referenced by `d2i_RSA_PUBKEY()`, `debug_print_crt()`, `generate_RSA_keys_ciphertext()`, `generate_RSA_keys_plaintext()`, `main()`, `rsa_check_privkey()`, `rsa_check_pubkey()`, `rsa_decryption()`, `rsa_encryption()`, `rsa_free()`, `rsa_gen_key()`, `rsa_public()`, `rsa_self_test()`, `x509parse_crt()`, and `x509parse_key()`.

### 12.96.2.5 `int(* rsa_context::f_rng)(void *)`

RNG function

Referenced by `rsa_gen_key()`, and `rsa_init()`.

### 12.96.2.6 `int rsa_context::hash_id`

hash identifier

Definition at line 76 of file `rsa.h`.

Referenced by `rsa_init()`.

### 12.96.2.7 `int rsa_context::len`

size(N) in chars

Definition at line 59 of file `rsa.h`.

Referenced by `d2i_RSA_PUBKEY()`, `main()`, `rsa_decryption()`, `rsa_encryption()`, `rsa_gen_key()`, `rsa_pkcs1_decrypt()`, `rsa_pkcs1_encrypt()`, `rsa_pkcs1_sign()`, `rsa_pkcs1_verify()`, `rsa_private()`, `rsa_public()`, `rsa_self_test()`, `ssl_parse_certificate_verify()`, `ssl_parse_client_key_exchange()`, `ssl_parse_server_key_exchange()`, `ssl_write_certificate_verify()`, `ssl_write_client_key_exchange()`, `ssl_write_server_key_exchange()`, `x509parse_crt()`, and `x509parse_key()`.

### 12.96.2.8 `mpi rsa_context::N`

public modulus

Definition at line 61 of file `rsa.h`.

Referenced by `d2i_RSA_PUBKEY()`, `debug_print_crt()`, `generate_RSA_keys_ciphertext()`, `generate_RSA_keys_plaintext()`, `main()`, `rsa_check_privkey()`, `rsa_check_pubkey()`, `rsa_decryption()`, `rsa_encryption()`, `rsa_free()`, `rsa_gen_key()`, `rsa_private()`, `rsa_public()`, `rsa_self_test()`, `x509parse_cert_info()`, `x509parse_crt()`, and `x509parse_key()`.

**12.96.2.9 mpi rsa\_context::P**

1st prime factor

Definition at line 65 of file rsa.h.

Referenced by generate\_RSA\_keys\_ciphertext(), generate\_RSA\_keys\_plaintext(), main(), rsa\_check\_privkey(), rsa\_decryption(), rsa\_free(), rsa\_gen\_key(), rsa\_private(), rsa\_self\_test(), and x509parse\_key().

**12.96.2.10 void\* rsa\_context::p\_rng**

RNG parameter

Definition at line 78 of file rsa.h.

Referenced by rsa\_gen\_key(), and rsa\_init().

**12.96.2.11 int rsa\_context::padding**

1.5 or OAEP/PSS

Definition at line 75 of file rsa.h.

Referenced by rsa\_init(), rsa\_pkcs1\_decrypt(), rsa\_pkcs1\_encrypt(), rsa\_pkcs1\_sign(), and rsa\_pkcs1\_verify().

**12.96.2.12 mpi rsa\_context::Q**

2nd prime factor

Definition at line 66 of file rsa.h.

Referenced by generate\_RSA\_keys\_ciphertext(), generate\_RSA\_keys\_plaintext(), main(), rsa\_check\_privkey(), rsa\_decryption(), rsa\_free(), rsa\_gen\_key(), rsa\_private(), rsa\_self\_test(), and x509parse\_key().

**12.96.2.13 mpi rsa\_context::QP**

$1 / (Q \% P)$

Definition at line 69 of file rsa.h.

Referenced by generate\_RSA\_keys\_ciphertext(), generate\_RSA\_keys\_plaintext(), main(), rsa\_decryption(), rsa\_free(), rsa\_gen\_key(), rsa\_private(), rsa\_self\_test(), and x509parse\_key().

**12.96.2.14 mpi rsa\_context::RN**

cached  $R^2 \bmod N$

Definition at line 71 of file rsa.h.

Referenced by rsa\_free(), rsa\_private(), and rsa\_public().

**12.96.2.15 mpi rsa\_context::RP**

cached  $R^2 \bmod P$

Definition at line 72 of file rsa.h.

Referenced by `rsa_free()`, and `rsa_private()`.

#### **12.96.2.16   `mpi rsa_context::RQ`**

cached  $R^2 \bmod Q$

Definition at line 73 of file rsa.h.

Referenced by `rsa_free()`, and `rsa_private()`.

#### **12.96.2.17   `int rsa_context::ver`**

always 0

Definition at line 58 of file rsa.h.

Referenced by `x509parse_key()`.

The documentation for this struct was generated from the following file:

- `/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/rsa.h`

## 12.97 LibMC::Properties::Settings Class Reference

### Public Member Functions

- [Settings](#) ()

### Properties

- static [Settings Default](#) [get]

### Private Member Functions

- void [SettingChangingEventHandler](#) (object sender, System.Configuration.SettingChangingEventArgs e)
- void [SettingsSavingEventHandler](#) (object sender, System.ComponentModel.CancelEventArgs e)

### Static Private Attributes

- static [Settings defaultInstance](#) = (([Settings](#))(global::System.Configuration.ApplicationSettingsBase.Synchronized(new [Settings](#)())))

#### 12.97.1 Detailed Description

Definition at line 16 of file Settings.Designer.cs.

#### 12.97.2 Constructor & Destructor Documentation

##### 12.97.2.1 LibMC::Properties::Settings::Settings () [inline]

Definition at line 19 of file Settings.cs.

#### 12.97.3 Member Function Documentation

##### 12.97.3.1 void LibMC::Properties::Settings::SettingChangingEventHandler (object sender, System.Configuration.SettingChangingEventArgs e) [inline, private]

Definition at line 28 of file Settings.cs.

##### 12.97.3.2 void LibMC::Properties::Settings::SettingsSavingEventHandler (object sender, System.ComponentModel.CancelEventArgs e) [inline, private]

Definition at line 32 of file Settings.cs.

## 12.97.4 Field Documentation

**12.97.4.1 Settings LibMC::Properties::Settings::defaultInstance =**  
**((Settings)(global::System.Configuration.ApplicationSettingsBase.Synchronized(new**  
**Settings())) [static, private]**

Definition at line 18 of file Settings.Designer.cs.

## 12.97.5 Property Documentation

**12.97.5.1 Settings LibMC::Properties::Settings::Default [static, get]**

Definition at line 20 of file Settings.Designer.cs.

The documentation for this class was generated from the following files:

- [/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/win32/LibMC.net/LibMC/Properties/Settings.Designer.cs](#)
- [/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/win32/LibMC.net/LibMC/Settings.cs](#)

## 12.98 sha1\_context Struct Reference

SHA-1 context structure.

```
#include <sha1.h>
```

### Data Fields

- unsigned long [total](#) [2]
- unsigned long [state](#) [5]
- unsigned char [buffer](#) [64]
- unsigned char [ipad](#) [64]
- unsigned char [opad](#) [64]

### 12.98.1 Detailed Description

SHA-1 context structure.

Definition at line 10 of file `sha1.h`.

### 12.98.2 Field Documentation

#### 12.98.2.1 unsigned char sha1\_context::buffer[64]

data block being processed

Definition at line 14 of file `sha1.h`.

Referenced by `sha1_update()`.

#### 12.98.2.2 unsigned char sha1\_context::ipad[64]

HMAC: inner padding

Definition at line 16 of file `sha1.h`.

Referenced by `sha1_hmac_starts()`.

#### 12.98.2.3 unsigned char sha1\_context::opad[64]

HMAC: outer padding

Definition at line 17 of file `sha1.h`.

Referenced by `sha1_hmac_finish()`, and `sha1_hmac_starts()`.

#### 12.98.2.4 unsigned long sha1\_context::state[5]

intermediate digest state

Definition at line 13 of file `sha1.h`.

Referenced by `sha1_finish()`, `sha1_process()`, `sha1_starts()`, and `ssl_calc_finished()`.

**12.98.2.5 unsigned long sha1\_context::total[2]**

number of bytes processed

Definition at line 12 of file sha1.h.

Referenced by sha1\_finish(), sha1\_starts(), and sha1\_update().

The documentation for this struct was generated from the following file:

- </home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/sha1.h>



## 12.99 sha2\_context Struct Reference

SHA-256 context structure.

```
#include <sha2.h>
```

### Data Fields

- unsigned long [total](#) [2]
- unsigned long [state](#) [8]
- unsigned char [buffer](#) [64]
- unsigned char [ipad](#) [64]
- unsigned char [opad](#) [64]
- [int is224](#)

### 12.99.1 Detailed Description

SHA-256 context structure.

Definition at line 10 of file sha2.h.

### 12.99.2 Field Documentation

#### 12.99.2.1 unsigned char sha2\_context::buffer[64]

data block being processed

Definition at line 14 of file sha2.h.

Referenced by sha2\_update().

#### 12.99.2.2 unsigned char sha2\_context::ipad[64]

HMAC: inner padding

Definition at line 16 of file sha2.h.

Referenced by sha2\_hmac\_starts().

#### 12.99.2.3 int sha2\_context::is224

0 => SHA-256, else SHA-224

Definition at line 18 of file sha2.h.

Referenced by sha2\_finish(), sha2\_hmac\_finish(), and sha2\_starts().

#### 12.99.2.4 unsigned char sha2\_context::opad[64]

HMAC: outer padding

Definition at line 17 of file sha2.h.

Referenced by sha2\_hmac\_finish(), and sha2\_hmac\_starts().

**12.99.2.5 unsigned long sha2\_context::state[8]**

intermediate digest state

Definition at line 13 of file sha2.h.

Referenced by sha2\_finish(), sha2\_process(), and sha2\_starts().

**12.99.2.6 unsigned long sha2\_context::total[2]**

number of bytes processed

Definition at line 12 of file sha2.h.

Referenced by sha2\_finish(), sha2\_starts(), and sha2\_update().

The documentation for this struct was generated from the following file:

- [/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/sha2.h](#)

## 12.100 sha4\_context Struct Reference

SHA-512 context structure.

```
#include <sha4.h>
```

### Data Fields

- unsigned int64 [total](#) [2]
- unsigned int64 [state](#) [8]
- unsigned char [buffer](#) [128]
- unsigned char [ipad](#) [128]
- unsigned char [opad](#) [128]
- [int is384](#)

### 12.100.1 Detailed Description

SHA-512 context structure.

Definition at line 18 of file sha4.h.

### 12.100.2 Field Documentation

#### 12.100.2.1 unsigned char sha4\_context::buffer[128]

data block being processed

Definition at line 22 of file sha4.h.

Referenced by sha4\_update().

#### 12.100.2.2 unsigned char sha4\_context::ipad[128]

HMAC: inner padding

Definition at line 24 of file sha4.h.

Referenced by sha4\_hmac\_starts().

#### 12.100.2.3 int sha4\_context::is384

0 => SHA-512, else SHA-384

Definition at line 26 of file sha4.h.

Referenced by sha4\_finish(), sha4\_hmac\_finish(), and sha4\_starts().

#### 12.100.2.4 unsigned char sha4\_context::opad[128]

HMAC: outer padding

Definition at line 25 of file sha4.h.

Referenced by sha4\_hmac\_finish(), and sha4\_hmac\_starts().

**12.100.2.5   unsigned int64 sha4\_context::state[8]**

intermediate digest state

Definition at line 21 of file sha4.h.

Referenced by sha4\_finish(), sha4\_process(), and sha4\_starts().

**12.100.2.6   unsigned int64 sha4\_context::total[2]**

number of bytes processed

Definition at line 20 of file sha4.h.

Referenced by sha4\_finish(), sha4\_starts(), and sha4\_update().

The documentation for this struct was generated from the following file:

- [/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/sha4.h](#)

## 12.101 syncList\_s Struct Reference

```
#include <sync_list.h>
```

### Data Fields

- `RWLOCK_T * lock`
- `MUTEX_T * giant_lock`
- `list_p list`
- `int size`

### 12.101.1 Detailed Description

Definition at line 26 of file `sync_list.h`.

### 12.101.2 Field Documentation

#### 12.101.2.1 `MUTEX_T* syncList_s::giant_lock`

Definition at line 28 of file `sync_list.h`.

Referenced by `MC_SyncDelete()`, and `MC_SyncInit()`.

#### 12.101.2.2 `list_p syncList_s::list`

Definition at line 30 of file `sync_list.h`.

Referenced by `syncListAddNode()`, `syncListDelete()`, `syncListFind()`, and `syncListRemove()`.

#### 12.101.2.3 `RWLOCK_T* syncList_s::lock`

Definition at line 27 of file `sync_list.h`.

Referenced by `syncListAddNode()`, `syncListDelete()`, `syncListFind()`, `syncListInit()`, and `syncListRemove()`.

#### 12.101.2.4 `int syncList_s::size`

Definition at line 31 of file `sync_list.h`.

The documentation for this struct was generated from the following file:

- `/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mc_sync/sync_list.h`

## 12.102 syncListNode\_s Struct Reference

```
#include <sync_list.h>
```

### Data Fields

- `MUTEX_T * lock`
- `COND_T * cond`
- `SEMAPHORE_T * sem`
- `int id`
- `int signalled`

### 12.102.1 Detailed Description

Definition at line 16 of file `sync_list.h`.

### 12.102.2 Field Documentation

#### 12.102.2.1 `COND_T* syncListNode_s::cond`

Definition at line 18 of file `sync_list.h`.

Referenced by `MC_CondBroadcast()`, `MC_CondSignal()`, `MC_CondWait()`, `syncListNodeDestroy()`, `syncListNodeInit()`, and `syncListNodeNew()`.

#### 12.102.2.2 `int syncListNode_s::id`

Definition at line 20 of file `sync_list.h`.

Referenced by `MC_SyncInit()`, `syncListAddNode()`, `syncListDelete()`, and `syncListRemove()`.

#### 12.102.2.3 `MUTEX_T* syncListNode_s::lock`

Definition at line 17 of file `sync_list.h`.

Referenced by `MC_CondBroadcast()`, `MC_CondReset()`, `MC_CondSignal()`, `MC_CondWait()`, `MC_MutexLock()`, `MC_MutexUnlock()`, `MC_SyncDelete()`, `syncListNodeDestroy()`, `syncListNodeInit()`, and `syncListNodeNew()`.

#### 12.102.2.4 `SEMAPHORE_T* syncListNode_s::sem`

Definition at line 19 of file `sync_list.h`.

Referenced by `MC_SemaphorePost()`, `MC_SemaphoreWait()`, `syncListNodeDestroy()`, `syncListNodeInit()`, and `syncListNodeNew()`.

#### 12.102.2.5 `int syncListNode_s::signalled`

Definition at line 21 of file `sync_list.h`.

Referenced by MC\_CondBroadcast(), MC\_CondReset(), MC\_CondSignal(), MC\_CondWait(), and syncListNodeNew().

The documentation for this struct was generated from the following file:

- [/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mc\\_sync/sync\\_list.h](/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mc_sync/sync_list.h)





## Chapter 13

# File Documentation

### 13.1 `/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/acc.c` File Reference

```
#include <sys/socket.h>
#include <arpa/inet.h>
#include <netinet/in.h>
#include <netdb.h>
#include <sys/un.h>
#include <unistd.h>
#include <sys/time.h>
#include <pthread.h>
#include "config.h"
#include <stdlib.h>
#include "include/acc.h"
#include "include/connection.h"
#include "include/data_structures.h"
#include "include/macros.h"
#include "include/mc_error.h"
#include "include/mc_platform.h"
#include "include/message.h"
#include "include/mtp_http.h"
#include "include/xml_parser.h"
#include "include/fipa_acl_envelope.h"
```

## Defines

- #define [BACKLOG](#) 200
- #define [CONN\\_THREADS](#) 40
- #define [CONNECT\\_THREAD\\_EXIT](#)()
- #define [BUFLen](#) 512
- #define [UDPPort](#) 8866

## Functions

- [acc\\_p acc\\_Initialize](#) (struct [mc\\_platform\\_s](#) \*[mc\\_platform](#))
- [int acc\\_Destroy](#) ([acc\\_p](#) [acc](#))
- [void \\* acc\\_MessageHandlerThread](#) ([void](#) \*[arg](#))
- [void \\* acc\\_Thread](#) ([void](#) \*[arg](#))
- [void \\* acc\\_connection\\_Thread](#) ([void](#) \*[arg](#))
- [void acc\\_Start](#) ([mc\\_platform\\_p](#) [mc\\_platform](#))
- [int auth\\_conn\\_rece\\_key](#) ([int](#) [sockfd](#), [char](#) \*[peer\\_name](#), [int](#) \*[nonce](#), [unsigned char](#) \*[aes\\_key](#), [char](#) \*[privkey](#), [char](#) \*[known\\_host\\_filename](#))
- [void \\* listen\\_Thread](#) ([void](#) \*[arg](#))
- [void \\* udplisten\\_Thread](#) ([void](#) \*[arg](#))

### 13.1.1 Define Documentation

#### 13.1.1.1 #define BACKLOG 200

Definition at line 65 of file [acc.c](#).

Referenced by [listen\\_Thread](#)() .

#### 13.1.1.2 #define BUFLen 512

Definition at line 788 of file [acc.c](#).

Referenced by [udplisten\\_Thread](#)() .

#### 13.1.1.3 #define CONN\_THREADS 40

Definition at line 284 of file [acc.c](#).

Referenced by [acc\\_Thread](#)() .

#### 13.1.1.4 #define CONNECT\_THREAD\_EXIT()

##### Value:

```
free(arg); \
MUTEX_LOCK(&acc->conn_thread_lock); \
acc->num_conn_threads--; \
COND_SIGNAL(&acc->conn_thread_cond); \
MUTEX_UNLOCK(&acc->conn_thread_lock); \
THREAD_EXIT();
```

Definition at line 378 of file acc.c.

Referenced by acc\_connection\_Thread().

#### 13.1.1.5 #define UDPPORT 8866

Definition at line 789 of file acc.c.

Referenced by udplisten\_Thread().

### 13.1.2 Function Documentation

#### 13.1.2.1 void\* acc\_connection\_Thread (void \* arg)

Definition at line 388 of file acc.c.

References mc\_platform\_s::acc, fipa\_agent\_identifier\_s::addresses, agent\_mailbox\_Post(), mc\_platform\_s::agent\_queue, AGENT\_UPDATE, CANCEL, CONNECT\_THREAD\_EXIT, connection\_Destroy(), mtp\_http\_s::content, mtp\_http\_content\_s::data, FIPA\_ACL, fipa\_acl\_envelope\_Destroy(), fipa\_acl\_envelope\_New(), fipa\_acl\_message\_Destroy(), fipa\_acl\_message\_New(), fipa\_acl\_Parse(), fipa\_agent\_identifier\_set\_s::fipa\_agent\_identifiers, fipa\_envelope\_Parse(), fipa\_message\_string\_Destroy(), fipa\_message\_string\_New(), mtp\_http\_s::http\_performative, HTTP\_POST, HTTP\_PUT, agent\_s::mailbox, agent\_s::mc\_platform, mc\_platform, fipa\_message\_string\_s::message, message\_s::message\_body, message\_Destroy(), message\_New(), mtp\_http\_s::message\_parts, mc\_platform\_s::message\_queue, message\_s::message\_type, message\_xml\_parse(), MOBILE\_AGENT, mtp\_http\_Destroy(), mtp\_http\_InitializeFromConnection(), mtp\_http\_New(), MXML\_NO\_CALLBACK, mxmlloadString(), N\_UNDRSTD, fipa\_agent\_identifier\_s::name, fipa\_agent\_identifier\_set\_s::num, fipa\_acl\_envelope\_s::num\_params, fipa\_acl\_envelope\_s::params, fipa\_message\_string\_s::parse, mc\_platform\_s::port, mc\_platform\_s::private\_key, QUER\_IF, QUER\_REF, RELAY, REQUEST, RETURN\_MSG, fipa\_url\_s::str, SUBSCRIBE, mtp\_http\_s::target, fipa\_acl\_Param\_s::to, fipa\_url\_sequence\_s::urls, and message\_s::xml\_root.

Referenced by acc\_Thread().

#### 13.1.2.2 int acc\_Destroy (acc\_p acc)

Definition at line 93 of file acc.c.

References MC\_SUCCESS.

Referenced by mc\_platform\_Destroy().

#### 13.1.2.3 acc\_p acc\_Initialize (struct mc\_platform\_s \* mc\_platform)

Definition at line 68 of file acc.c.

References COND\_INIT, COND\_T, MUTEX\_INIT, and MUTEX\_T.

Referenced by mc\_platform\_Initialize().

#### 13.1.2.4 void\* acc\_MessageHandlerThread (void \* arg)

Definition at line 105 of file acc.c.

References `mc_platform_s::acc`, `agent_Initialize()`, `mc_platform_s::agent_queue`, `agent_s::agent_status`, `AGENT_UPDATE`, `mc_platform_s::ams`, `CANCEL`, `COND_BROADCAST`, `COND_WAIT`, `agent_s::datastate`, `FIPA_ACL`, `mc_platform_s::giant`, `mc_platform_s::giant_cond`, `mc_platform_s::giant_lock`, `agent_s::lock`, `MC_AGENT_NEUTRAL`, `mc_platform`, `MC_RECV_AGENT`, `MC_RECV_MESSAGE`, `MC_RECV_RETURN`, `mc_platform_s::MC_signal`, `mc_platform_s::MC_signal_cond`, `mc_platform_s::MC_signal_lock`, `message_Destroy()`, `mc_platform_s::message_queue`, `message_Send()`, `message_s::message_type`, `MOBILE_AGENT`, `MUTEX_LOCK`, `MUTEX_UNLOCK`, `N_UNDRSTD`, `agent_s::name`, `agent_datastate_s::persistent`, `QUER_IF`, `QUER_REF`, `mc_platform_s::quit`, `mc_platform_s::quit_lock`, `RELAY`, `REQUEST`, `RETURN_MSG`, `SUBSCRIBE`, `THREAD_EXIT`, and `message_s::to_address`.

Referenced by `acc_Start()`.

### 13.1.2.5 void acc\_Start (mc\_platform\_p mc\_platform)

Definition at line 564 of file `acc.c`.

References `mc_platform_s::acc`, `acc_MessageHandlerThread()`, `acc_Thread()`, `listen_Thread()`, `MC_THREAD_ACC`, `mc_platform_s::stack_size`, `THREAD_CREATE`, and `udplisten_Thread()`.

Referenced by `mc_platform_Initialize()`.

### 13.1.2.6 void\* acc\_Thread (void \* arg)

Definition at line 287 of file `acc.c`.

References `mc_platform_s::acc`, `acc_connection_Thread()`, `COND_BROADCAST`, `COND_WAIT`, `CONN_THREADS`, `mc_platform_s::connection_queue`, `mc_platform_s::giant`, `mc_platform_s::giant_cond`, `mc_platform_s::giant_lock`, `mc_platform`, `MC_RECV_CONNECTION`, `mc_platform_s::MC_signal`, `mc_platform_s::MC_signal_cond`, `mc_platform_s::MC_signal_lock`, `MUTEX_LOCK`, `MUTEX_UNLOCK`, `mc_platform_s::quit`, `mc_platform_s::quit_lock`, `THREAD_CREATE`, `THREAD_DETACH`, `THREAD_EXIT`, and `THREAD_T`.

Referenced by `acc_Start()`.

### 13.1.2.7 int auth\_conn\_rece\_key (int sockfd, char \* peer\_name, int \* nonce, unsigned char \* aes\_key, char \* privkey, char \* known\_host\_filename)

Definition at line 613 of file `acc.c`.

References `read_known_host_file()`, and `reply_migration_process()`.

Referenced by `listen_Thread()`.

### 13.1.2.8 void\* listen\_Thread (void \* arg)

Definition at line 652 of file `acc.c`.

References `mc_platform_s::acc`, `connection_s::addr`, `connection_s::AES_key`, `mc_platform_s::agency`, `auth_conn_rece_key()`, `BACKLOG`, `connection_s::clientfd`, `COND_BROADCAST`, `connection_s::connect_id`, `connection_New()`, `mc_platform_s::connection_queue`, `agency_s::known_host_filename`, `mc_platform`, `MUTEX_LOCK`, `MUTEX_UNLOCK`, `connection_s::nonce`, `mc_platform_s::port`, `mc_platform_s::private_key`, `connection_s::remote_hostname`, `connection_s::serverfd`, `SOCKET_ERROR`, `mc_platform_s::sockfd`, and `THREAD_EXIT`.

Referenced by `acc_Start()`.

**13.1.2.9 void\* udplisten\_Thread (void \* *arg*)**

Definition at line 792 of file acc.c.

References `buf`, `BUFLen`, `mc_platform_s::hostname`, `mc_platform`, `PACKAGE_VERSION`, `mc_platform_s::port`, and `UDPPORT`.

Referenced by `acc_Start()`.

## 13.2 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/agent.c File Reference

```
#include <unistd.h>
#include "config.h"
#include <embedch.h>
#include "include/libmc.h"
#include "include/agent.h"
#include "include/mc_platform.h"
#include "include/message.h"
#include "include/agent_lib.h"
#include "include/interpreter_variable_data.h"
#include "include/xml_parser.h"
```

### Functions

- [int agent\\_AddPersistentVariable](#) ([agent\\_p](#) agent, [int](#) task\_num, const char \*var\_name)
- [agent\\_p](#) agent\_Copy (const [agent\\_p](#) agent)
- [agent\\_p](#) agent\_New (void)
- [agent\\_p](#) agent\_NewBinary (struct [mc\\_platform\\_s](#) \*mc\_platform)
- [agent\\_p](#) agent\_Initialize (struct [mc\\_platform\\_s](#) \*mc\_platform, [message\\_p](#) message, [int](#) id)
- [int](#) agent\_Destroy ([agent\\_p](#) agent)
- void [agent\\_RunChScript](#) ([agent\\_p](#) agent, [mc\\_platform\\_p](#) mc\_platform)
- void \* [agent\\_ChScriptInitVar](#) ([ChInterp\\_t](#) \*interp)
- void \* [agent\\_RunChScriptThread](#) (void \*ChAgent)

### 13.2.1 Function Documentation

#### 13.2.1.1 int agent\_AddPersistentVariable (agent\_p agent, int task\_num, const char \* var\_name)

Definition at line 52 of file agent.c.

References [agent\\_s::agent\\_interp](#), [agent\\_task\\_s::agent\\_variable\\_list](#), [interpreter\\_variable\\_data\\_s::array\\_dim](#), [interpreter\\_variable\\_data\\_s::array\\_extent](#), [CH\\_DATATYPE\\_SIZE](#), [CHECK\\_NULL](#), [interpreter\\_variable\\_data\\_s::data](#), [interpreter\\_variable\\_data\\_s::data\\_type](#), [agent\\_s::datastate](#), [MC\\_ERR](#), [agent\\_s::name](#), [interpreter\\_variable\\_data\\_s::name](#), [interpreter\\_variable\\_data\\_s::size](#), [size](#), [agent\\_datastate\\_s::task\\_progress](#), and [agent\\_datastate\\_s::tasks](#).

#### 13.2.1.2 void\* agent\_ChScriptInitVar (ChInterp\_t \* interp)

Definition at line 492 of file agent.c.

References [MC\\_AclAddReceiver\\_chdl\(\)](#), [MC\\_AclAddReplyTo\\_chdl\(\)](#), [MC\\_AclDestroy\\_chdl\(\)](#), [MC\\_AclNew\\_chdl\(\)](#), [MC\\_AclPost\\_chdl\(\)](#), [MC\\_AclReply\\_chdl\(\)](#), [MC\\_AclRetrieve\\_chdl\(\)](#), [MC\\_AclSend\\_chdl\(\)](#), [MC\\_AclSetContent\\_chdl\(\)](#), [MC\\_AclSetConversationID\\_chdl\(\)](#), [MC\\_AclSetPerformative\\_chdl\(\)](#), [MC\\_AclSetProtocol\\_chdl\(\)](#), [MC\\_AclSetSender\\_chdl\(\)](#), [MC\\_AclWaitRetrieve\\_chdl\(\)](#), [MC\\_AddAgent\\_chdl\(\)](#), [MC\\_AgentVariableRetrieve\\_chdl\(\)](#), [MC\\_AgentVariableSave\\_chdl\(\)](#), [MC\\_](#)

Barrier\_chdl(), MC\_BarrierDelete\_chdl(), MC\_BarrierInit\_chdl(), MC\_CallAgentFunc\_chdl(), MC\_ComposeAgent\_chdl(), MC\_ComposeAgentS\_chdl(), MC\_CondBroadcast\_chdl(), MC\_CondReset\_chdl(), MC\_CondSignal\_chdl(), MC\_CondWait\_chdl(), MC\_DeleteAgent\_chdl(), MC\_DeleteAgentWG\_chdl(), MC\_DeregisterService\_chdl(), MC\_DestroyServiceSearchResult\_chdl(), MC\_End\_chdl(), MC\_FindAgentByID\_chdl(), MC\_FindAgentByName\_chdl(), MC\_GetAgentID\_chdl(), MC\_GetAgentName\_chdl(), MC\_GetAgentStatus\_chdl(), MC\_GetAgentXMLString\_chdl(), MC\_GetTimeOfDay\_chdl(), MC\_HaltAgency\_chdl(), MC\_MigrateAgent\_chdl(), MC\_MutexLock\_chdl(), MC\_MutexUnlock\_chdl(), MC\_PrintAgentCode\_chdl(), MC\_RegisterService\_chdl(), MC\_ResumeAgency\_chdl(), MC\_RetrieveAgent\_chdl(), MC\_RetrieveAgentCode\_chdl(), MC\_SaveData\_chdl(), MC\_SearchForService\_chdl(), MC\_SemaphorePost\_chdl(), MC\_SemaphoreWait\_chdl(), MC\_SendAgentMigrationMessage\_chdl(), MC\_SendAgentMigrationMessageFile\_chdl(), MC\_SendSteerCommand\_chdl(), MC\_SetAgentStatus\_chdl(), MC\_SetDefaultAgentStatus\_chdl(), MC\_SyncDelete\_chdl(), MC\_SyncInit\_chdl(), MC\_TerminateAgent\_chdl(), and MC\_TerminateAgentWG\_chdl().

Referenced by AP\_QUEUE\_STD\_DEFN\_TEMPLATE(), and mc\_platform\_Initialize().

### 13.2.1.3 agent\_p agent\_Copy (const agent\_p agent)

Definition at line 134 of file agent.c.

References agent\_datastate\_Copy(), agent\_s::agent\_interp, agent\_mailbox\_New(), agent\_s::agent\_persistent, agent\_s::agent\_status, agent\_s::agent\_type, agent\_s::arrival\_time, agent\_s::datastate, agent\_s::home, agent\_s::home\_port, agent\_s::id, agent\_s::lock, agent\_s::mailbox, MUTEX\_INIT, MUTEX\_LOCK, MUTEX\_T, agent\_s::name, agent\_s::orphan, agent\_s::owner, agent\_s::return\_data, and agent\_s::run\_lock.

Referenced by MC\_CopyAgent().

### 13.2.1.4 int agent\_Destroy (agent\_p agent)

Definition at line 414 of file agent.c.

References agent\_datastate\_Destroy(), agent\_s::agent\_interp, agent\_mailbox\_Destroy(), agent\_s::agent\_status, agent\_s::datastate, agent\_s::home, mc\_platform\_s::interpreter\_queue, agent\_s::lock, agent\_s::mailbox, MC\_AGENT\_NEUTRAL, agent\_s::mc\_platform, MC\_SUCCESS, MUTEX\_DESTROY, MUTEX\_LOCK, agent\_s::name, agent\_s::owner, agent\_s::run\_lock, agent\_s::sender, and agent\_s::wg\_code.

Referenced by agent\_Initialize().

### 13.2.1.5 agent\_p agent\_Initialize (struct mc\_platform\_s \* mc\_platform, message\_p message, int id)

Definition at line 293 of file agent.c.

References agent\_s::agent\_address, agent\_datastate\_New(), agent\_Destroy(), agent\_mailbox\_New(), agent\_s::agent\_pipe\_active, agent\_s::agent\_pipe\_ready\_to\_read, agent\_s::agent\_ready\_to\_send, agent\_s::agent\_script\_ready, agent\_s::agent\_status, agent\_s::agent\_thread\_id, agent\_s::agent\_type, message\_s::agent\_xml\_flag, agent\_xml\_parse(), agent\_s::arrival\_time, agent\_s::datastate, mc\_platform\_s::default\_agentstatus, mc\_platform\_s::err, agent\_s::home, mc\_platform\_s::hostname, agent\_s::id, agent\_s::lock, agent\_s::mailbox, MC\_ERR\_PARSE, agent\_s::mc\_platform, MC\_REMOTE\_AGENT, MC\_RETURN\_AGENT, MC\_WAIT\_CH, message\_s::message\_type, MOBILE\_AGENT, MUTEX\_DESTROY, MUTEX\_INIT, MUTEX\_T, agent\_s::orphan, mc\_platform\_s::port, RETURN\_MSG,

agent\_s::run\_lock, agent\_s::sender, agent\_datastate\_s::xml\_agent\_root, message\_s::xml\_payload, message\_s::xml\_root, and agent\_datastate\_s::xml\_root.

Referenced by acc\_MessageHandlerThread(), and MC\_SendAgentMigrationMessageFile().

### 13.2.1.6 agent\_p agent\_New (void)

Definition at line 197 of file agent.c.

References agent\_s::lock, MUTEX\_INIT, MUTEX\_NEW, and agent\_s::run\_lock.

Referenced by MC\_ComposeAgentS().

### 13.2.1.7 agent\_p agent\_NewBinary (struct mc\_platform\_s \* mc\_platform)

Definition at line 220 of file agent.c.

References agent\_s::agent\_address, agent\_mailbox\_New(), agent\_s::agent\_pipe\_active, agent\_s::agent\_pipe\_ready\_to\_read, agent\_s::agent\_ready\_to\_send, agent\_s::agent\_script\_ready, agent\_s::agent\_status, agent\_s::agent\_thread\_id, agent\_s::arrival\_time, agent\_s::binary, agent\_s::home, mc\_platform\_s::hostname, agent\_s::id, agent\_s::lock, agent\_s::mailbox, MC\_AGENT\_ACTIVE, agent\_s::mc\_platform, MUTEX\_INIT, MUTEX\_T, agent\_s::orphan, mc\_platform\_s::port, agent\_s::run\_lock, and agent\_s::sender.

Referenced by MC\_AddStationaryAgent().

### 13.2.1.8 void agent\_RunChScript (agent\_p agent, mc\_platform\_p mc\_platform)

Definition at line 454 of file agent.c.

References agent\_RunChScriptThread(), agent\_s::agent\_status, agent\_s::agent\_thread, MC\_AGENT\_ACTIVE, agent\_s::mc\_platform, MC\_THREAD\_AGENT, mc\_platform\_s::stack\_size, and THREAD\_CREATE.

Referenced by ams\_ManageAgentList().

### 13.2.1.9 void\* agent\_RunChScriptThread (void \* ChAgent)

Definition at line 926 of file agent.c.

References agent\_s::agent\_interp, mc\_platform\_s::ams, COND\_SIGNAL, agent\_s::datastate, agent\_s::id, mc\_platform\_s::interp\_options, mc\_platform\_s::interpreter\_queue, interpreter\_queue\_CreateRetrieve(), interpreter\_variable\_data\_Destroy(), interpreter\_variable\_data\_Initialize(), interpreter\_variable\_data\_InitializeFromAgent(), MC\_AGENT\_NEUTRAL, MC\_EXEC\_AGENT, agent\_s::mc\_platform, mc\_platform, MC\_RETURN\_AGENT, mc\_platform\_s::MC\_signal, mc\_platform\_s::MC\_signal\_cond, mc\_platform\_s::MC\_signal\_lock, MC\_WAIT\_FINISHED, MC\_WAIT\_MESSGSEND, MUTEX\_LOCK, MUTEX\_UNLOCK, agent\_s::name, SIGNAL, and agent\_datastate\_s::task\_progress.

Referenced by agent\_RunChScript().



## 13.3 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/agent\_datastate.c File Reference

```
#include "config.h"
#include <mxml.h>
#include "include/agent_datastate.h"
#include "include/macros.h"
#include "include/mc_error.h"
```

### Functions

- [agent\\_datastate\\_p agent\\_datastate\\_Copy](#) (const [agent\\_datastate\\_p](#) datastate)
- [agent\\_datastate\\_p agent\\_datastate\\_New](#) (void)
- [int agent\\_datastate\\_Destroy](#) ([agent\\_datastate\\_p](#) agent\_datastate)

#### 13.3.1 Function Documentation

##### 13.3.1.1 [agent\\_datastate\\_p agent\\_datastate\\_Copy](#) (const [agent\\_datastate\\_p](#) datastate)

Definition at line 47 of file `agent_datastate.c`.

References `agent_datastate_s::agent_code`, `agent_datastate_s::agent_code_ids`, `agent_datastate_s::agent_codes`, `agent_datastate_New()`, `agent_task_Copy()`, `agent_datastate_s::init_agent_status`, `agent_datastate_s::number_of_tasks`, `agent_datastate_s::persistent`, `agent_datastate_s::return_data`, `agent_datastate_s::task_progress`, and `agent_datastate_s::tasks`.

Referenced by `agent_Copy()`.

##### 13.3.1.2 [int agent\\_datastate\\_Destroy](#) ([agent\\_datastate\\_p](#) agent\_datastate)

Definition at line 136 of file `agent_datastate.c`.

References `agent_datastate_s::agent_code_ids`, `agent_datastate_s::agent_codes`, `agent_task_Destroy()`, `MC_SUCCESS`, `mxmlDelete()`, `agent_datastate_s::number_of_tasks`, `agent_datastate_s::tasks`, and `agent_datastate_s::xml_root`.

Referenced by `agent_Destroy()`.

##### 13.3.1.3 [agent\\_datastate\\_p agent\\_datastate\\_New](#) (void)

Definition at line 115 of file `agent_datastate.c`.

References `agent_datastate_s::agent_code`, `CHECK_NULL`, `agent_datastate_s::init_agent_status`, `agent_datastate_s::number_of_tasks`, `agent_datastate_s::persistent`, `agent_datastate_s::progress_modifier`, `agent_datastate_s::return_data`, `agent_datastate_s::task_progress`, `agent_datastate_s::tasks`, `agent_datastate_s::xml_agent_root`, and `agent_datastate_s::xml_root`.

Referenced by `agent_datastate_Copy()`, `agent_Initialize()`, and `MC_ComposeAgentS()`.

## 13.4 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/agent\_mailbox.c File Reference

```
#include "config.h"
#include "include/data_structures.h"
```

### Functions

- [agent\\_mailbox\\_p agent\\_mailbox\\_New](#) (void)
- [agent\\_mailbox\\_p agent\\_mailbox\\_Copy](#) (agent\_mailbox\_p src)
- [int agent\\_mailbox\\_Destroy](#) (agent\_mailbox\_t \*mailbox)
- [int agent\\_mailbox\\_Post](#) (agent\_mailbox\_p mailbox, fipa\_acl\_message\_t \*message)
- [fipa\\_acl\\_message\\_t \\* agent\\_mailbox\\_Retrieve](#) (agent\_mailbox\_p mailbox)
- [fipa\\_acl\\_message\\_t \\* agent\\_mailbox\\_WaitRetrieve](#) (agent\_mailbox\_p mailbox)

### 13.4.1 Function Documentation

#### 13.4.1.1 agent\_mailbox\_p agent\_mailbox\_Copy (agent\_mailbox\_p src)

Definition at line 21 of file agent\_mailbox.c.

References agent\_mailbox\_New(), and agent\_mailbox\_s::mail\_queue.

#### 13.4.1.2 int agent\_mailbox\_Destroy (agent\_mailbox\_t \* mailbox)

Definition at line 29 of file agent\_mailbox.c.

References agent\_mailbox\_s::mail\_queue.

Referenced by agent\_Destroy().

#### 13.4.1.3 agent\_mailbox\_p agent\_mailbox\_New (void)

Definition at line 12 of file agent\_mailbox.c.

References agent\_mailbox\_s::mail\_queue.

Referenced by agent\_Copy(), agent\_Initialize(), agent\_mailbox\_Copy(), and agent\_NewBinary().

#### 13.4.1.4 int agent\_mailbox\_Post (agent\_mailbox\_p mailbox, fipa\_acl\_message\_t \* message)

Definition at line 38 of file agent\_mailbox.c.

References agent\_mailbox\_s::mail\_queue.

Referenced by acc\_connection\_Thread(), and MC\_AclPost().

#### 13.4.1.5 fipa\_acl\_message\_t\* agent\_mailbox\_Retrieve (agent\_mailbox\_p mailbox)

Definition at line 44 of file agent\_mailbox.c.

References agent\_mailbox\_s::mail\_queue.

Referenced by agent\_mailbox\_WaitRetrieve(), and MC\_AclRetrieve().

#### **13.4.1.6 fipa\_acl\_message\_t\* agent\_mailbox\_WaitRetrieve (agent\_mailbox\_p mailbox)**

Definition at line 49 of file agent\_mailbox.c.

References agent\_mailbox\_Retrieve(), COND\_WAIT, agent\_mailbox\_s::mail\_queue, MUTEX\_LOCK, and MUTEX\_UNLOCK.

Referenced by MC\_AclWaitRetrieve().

## 13.5 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/agent\_return\_data.c File Reference

```
#include "config.h"
#include "include/interpreter_variable_data.h"
#include "include/agent.h"
```

### Functions

- [interpreter\\_variable\\_data\\_p interpreter\\_variable\\_data\\_New](#) (void)
- [interpreter\\_variable\\_data\\_p interpreter\\_variable\\_data\\_InitializeFromAgent](#) ([agent\\_p](#) agent)
- [interpreter\\_variable\\_data\\_p interpreter\\_variable\\_data\\_Initialize](#) ([agent\\_p](#) agent, const char \*varname)
- [int interpreter\\_variable\\_data\\_Destroy](#) ([interpreter\\_variable\\_data\\_p](#) agent\_variable\_data)
- [interpreter\\_variable\\_data\\_p interpreter\\_variable\\_data\\_Copy](#) ([interpreter\\_variable\\_data\\_p](#) src)

### 13.5.1 Function Documentation

#### 13.5.1.1 [interpreter\\_variable\\_data\\_p interpreter\\_variable\\_data\\_Copy](#) ([interpreter\\_variable\\_data\\_p](#) src)

Definition at line 235 of file agent\_return\_data.c.

References [interpreter\\_variable\\_data\\_s::array\\_dim](#), [interpreter\\_variable\\_data\\_s::array\\_extent](#), [interpreter\\_variable\\_data\\_s::data](#), [interpreter\\_variable\\_data\\_s::data\\_type](#), [interpreter\\_variable\\_data\\_New\(\)](#), [interpreter\\_variable\\_data\\_s::name](#), and [interpreter\\_variable\\_data\\_s::size](#).

Referenced by [agent\\_task\\_Copy\(\)](#).

#### 13.5.1.2 [int interpreter\\_variable\\_data\\_Destroy](#) ([interpreter\\_variable\\_data\\_p](#) agent\_variable\_data)

Definition at line 216 of file agent\_return\_data.c.

References [interpreter\\_variable\\_data\\_s::array\\_extent](#), [interpreter\\_variable\\_data\\_s::data](#), [MC\\_SUCCESS](#), and [interpreter\\_variable\\_data\\_s::name](#).

Referenced by [agent\\_RunChScriptThread\(\)](#), and [agent\\_task\\_Destroy\(\)](#).

#### 13.5.1.3 [interpreter\\_variable\\_data\\_p interpreter\\_variable\\_data\\_Initialize](#) ([agent\\_p](#) agent, const char \* varname)

Definition at line 145 of file agent\_return\_data.c.

References [agent\\_s::agent\\_interp](#), [interpreter\\_variable\\_data\\_s::array\\_dim](#), [interpreter\\_variable\\_data\\_s::array\\_extent](#), [CH\\_DATATYPE\\_SIZE](#), [CHECK\\_NULL](#), [interpreter\\_variable\\_data\\_s::data](#), [interpreter\\_variable\\_data\\_s::data\\_type](#), [MUTEX\\_LOCK](#), [MUTEX\\_UNLOCK](#), [interpreter\\_variable\\_data\\_s::name](#), [agent\\_s::run\\_lock](#), [interpreter\\_variable\\_data\\_s::size](#), and [size](#).

Referenced by [agent\\_RunChScriptThread\(\)](#).

#### **13.5.1.4 interpreter\_variable\_data\_p interpreter\_variable\_data\_InitializeFromAgent (agent\_p agent)**

Definition at line 61 of file agent\_return\_data.c.

References agent\_s::agent\_interp, interpreter\_variable\_data\_s::array\_dim, interpreter\_variable\_data\_s::array\_extent, CH\_DATATYPE\_SIZE, CHECK\_NULL, interpreter\_variable\_data\_s::data, interpreter\_variable\_data\_s::data\_type, agent\_s::datastate, interpreter\_variable\_data\_s::name, interpreter\_variable\_data\_s::size, size, agent\_datastate\_s::task\_progress, agent\_datastate\_s::tasks, and agent\_task\_s::var\_name.

Referenced by agent\_RunChScriptThread().

#### **13.5.1.5 interpreter\_variable\_data\_p interpreter\_variable\_data\_New (void)**

Definition at line 46 of file agent\_return\_data.c.

References interpreter\_variable\_data\_s::array\_dim, interpreter\_variable\_data\_s::array\_extent, CHECK\_NULL, interpreter\_variable\_data\_s::data, interpreter\_variable\_data\_s::data\_type, interpreter\_variable\_data\_s::name, and interpreter\_variable\_data\_s::size.

Referenced by agent\_xml\_parse\_\_data(), interpreter\_variable\_data\_Copy(), and MC\_SaveData\_chdl().

## 13.6 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/agent\_task.c File Reference

```
#include "config.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "include/agent_task.h"
#include "include/mc_error.h"
```

### Functions

- [agent\\_task\\_p agent\\_task\\_New](#) (void)
- [agent\\_task\\_p agent\\_task\\_Copy](#) (agent\_task\_p task)
- [int agent\\_task\\_Destroy](#) (agent\_task\_p agent\_task)

#### 13.6.1 Function Documentation

##### 13.6.1.1 agent\_task\_p agent\_task\_Copy (agent\_task\_p task)

Definition at line 66 of file agent\_task.c.

References `agent_task_s::agent_return_data`, `agent_task_s::agent_variable_list`, `agent_task_s::code_id`, `agent_task_s::init_agent_status`, `interpreter_variable_data_Copy()`, `ListAdd()`, `ListSearch()`, `agent_task_s::num_saved_variables`, `agent_task_s::number_of_elements`, `agent_task_s::persistent`, `agent_task_s::saved_variables`, `agent_task_s::server_name`, `agent_task_s::size_of_element_array`, and `agent_task_s::var_name`.

Referenced by `agent_datastate_Copy()`.

##### 13.6.1.2 int agent\_task\_Destroy (agent\_task\_p agent\_task)

Definition at line 132 of file agent\_task.c.

References `agent_task_s::agent_return_data`, `agent_task_s::agent_variable_list`, `agent_task_s::code_id`, `interpreter_variable_data_Destroy()`, `MC_SUCCESS`, `agent_task_s::saved_variables`, `agent_task_s::server_name`, and `agent_task_s::var_name`.

Referenced by `agent_datastate_Destroy()`.

##### 13.6.1.3 agent\_task\_p agent\_task\_New (void)

Definition at line 48 of file agent\_task.c.

References `agent_task_s::agent_variable_list`, `agent_task_s::num_saved_variables`, and `agent_task_s::saved_variables`.

Referenced by `agent_xml_parse__tasks()`, and `MC_ComposeAgentS()`.

## 13.7 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/ams.c File Reference

```
#include "config.h"
#include "include/ams.h"
#include "include/agent.h"
#include "include/data_structures.h"
#include "include/mc_platform.h"
```

### Functions

- [int ams\\_Destroy](#) (ams\_p ams)
- [ams\\_p ams\\_Initialize](#) (mc\_platform\_p mc\_platform)
- [void ams\\_Print](#) (ams\_p ams)
- [int ams\\_ManageAgentList](#) (ams\_p ams)
- [void ams\\_Start](#) (mc\_platform\_p mc\_platform)
- [void \\* ams\\_Thread](#) (void \*arg)

### 13.7.1 Function Documentation

#### 13.7.1.1 int ams\_Destroy (ams\_p ams)

Definition at line 46 of file ams.c.

References COND\_DESTROY, MC\_SUCCESS, and MUTEX\_DESTROY.

Referenced by mc\_platform\_Destroy().

#### 13.7.1.2 ams\_p ams\_Initialize (mc\_platform\_p mc\_platform)

Definition at line 57 of file ams.c.

References CHECK\_NULL, COND\_INIT, COND\_T, MUTEX\_INIT, and MUTEX\_T.

Referenced by mc\_platform\_Initialize().

#### 13.7.1.3 int ams\_ManageAgentList (ams\_p ams)

Definition at line 116 of file ams.c.

References [agent\\_RunChScript\(\)](#), [agent\\_s::agent\\_status](#), [agent\\_s::binary](#), [ListSearch\(\)](#), [agent\\_s::lock](#), [MC\\_AGENT\\_ACTIVE](#), [MC\\_AGENT\\_NEUTRAL](#), [MC\\_TerminateAgent\(\)](#), [MC\\_WAIT\\_CH](#), [MC\\_WAIT\\_FINISHED](#), [MC\\_WAIT\\_MESSGSEND](#), [message\\_Destroy\(\)](#), [message\\_InitializeFromAgent\(\)](#), [message\\_New\(\)](#), [MUTEX\\_LOCK](#), [MUTEX\\_UNLOCK](#), [agent\\_s::name](#), [agent\\_s::orphan](#), [mc\\_platform\\_s::quit](#), [mc\\_platform\\_s::quit\\_lock](#), and [agent\\_s::run\\_lock](#).

Referenced by [ams\\_Thread\(\)](#).

**13.7.1.4 void ams\_Print (ams\_p *ams*)**

Definition at line 84 of file ams.c.

References agent\_s::agent\_status, agent\_s::connect\_id, agent\_s::id, ListSearch(), MUTEX\_LOCK, and MUTEX\_UNLOCK.

**13.7.1.5 void ams\_Start (mc\_platform\_p *mc\_platform*)**

Definition at line 223 of file ams.c.

References mc\_platform\_s::ams, ams\_Thread(), MC\_THREAD\_AMS, mc\_platform\_s::stack\_size, and THREAD\_CREATE.

Referenced by mc\_platform\_Initialize().

**13.7.1.6 void\* ams\_Thread (void \* *arg*)**

Definition at line 254 of file ams.c.

References mc\_platform\_s::ams, ams\_ManageAgentList(), COND\_BROADCAST, COND\_WAIT, mc\_platform, MUTEX\_LOCK, MUTEX\_UNLOCK, mc\_platform\_s::quit, mc\_platform\_s::quit\_lock, and THREAD\_EXIT.

Referenced by ams\_Start().



## 13.8 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/barrier.c File Reference

```
#include "config.h"
#include "include/barrier.h"
#include "include/mc_error.h"
```

### Functions

- [barrier\\_node\\_p barrier\\_node\\_Initialize](#) ([int id](#), [int num\\_registered](#))
- [int barrier\\_node\\_Destroy](#) ([barrier\\_node\\_p node](#))
- [int barrier\\_queue\\_Add](#) ([barrier\\_queue\\_p list](#), [barrier\\_node\\_p node](#))
- [int barrier\\_queue\\_Delete](#) ([int id](#), [barrier\\_queue\\_p list](#))
- [int barrier\\_queue\\_Destroy](#) ([barrier\\_queue\\_p queue](#))
- [barrier\\_node\\_p barrier\\_queue\\_Get](#) ([barrier\\_queue\\_p list](#), [int id](#))
- [barrier\\_queue\\_p barrier\\_queue\\_New](#) ([void](#))
- [barrier\\_node\\_p barrier\\_queue\\_Pop](#) ([barrier\\_queue\\_p queue](#))

### 13.8.1 Function Documentation

#### 13.8.1.1 [int barrier\\_node\\_Destroy](#) ([barrier\\_node\\_p node](#))

Definition at line 70 of file barrier.c.

References [barrier\\_node\\_s::cond](#), [COND\\_DESTROY](#), [barrier\\_node\\_s::lock](#), [MC\\_SUCCESS](#), and [MUTEX\\_DESTROY](#).

Referenced by [barrier\\_queue\\_Delete\(\)](#), and [barrier\\_queue\\_Destroy\(\)](#).

#### 13.8.1.2 [barrier\\_node\\_p barrier\\_node\\_Initialize](#) ([int id](#), [int num\\_registered](#))

Definition at line 45 of file barrier.c.

References [CHECK\\_NULL](#), [barrier\\_node\\_s::cond](#), [COND\\_INIT](#), [COND\\_T](#), [barrier\\_node\\_s::id](#), [barrier\\_node\\_s::lock](#), [MUTEX\\_INIT](#), [MUTEX\\_T](#), [node](#), [barrier\\_node\\_s::num\\_registered](#), and [barrier\\_node\\_s::num\\_waiting](#).

Referenced by [MC\\_BarrierInit\(\)](#).

#### 13.8.1.3 [int barrier\\_queue\\_Add](#) ([barrier\\_queue\\_p list](#), [barrier\\_node\\_p node](#))

Definition at line 87 of file barrier.c.

References [DATA](#), [barrier\\_node\\_s::id](#), [barrier\\_queue\\_s::list](#), [ListAdd\(\)](#), [list\\_s::listhead](#), [barrier\\_queue\\_s::lock](#), [MC\\_SUCCESS](#), [MC\\_WARN\\_DUPLICATE](#), [listNode\\_s::next](#), [listNode\\_s::node\\_data](#), [RWLOCK\\_WRLOCK](#), [RWLOCK\\_WRUNLOCK](#), and [barrier\\_queue\\_s::size](#).

Referenced by [MC\\_BarrierInit\(\)](#).

#### 13.8.1.4 `int barrier_queue_Delete (int id, barrier_queue_p list)`

Definition at line 111 of file barrier.c.

References `barrier_node_Destroy()`, `barrier_node_s::id`, `barrier_queue_s::list`, `ListDelete()`, `ListSearch()`, `barrier_queue_s::lock`, `MC_ERR_NOT_FOUND`, `MC_SUCCESS`, `RWLOCK_WRLOCK`, `RWLOCK_WRUNLOCK`, `barrier_queue_s::size`, and `list_s::size`.

Referenced by `MC_BarrierDelete()`.

#### 13.8.1.5 `int barrier_queue_Destroy (barrier_queue_p queue)`

Definition at line 131 of file barrier.c.

References `barrier_node_Destroy()`, `barrier_queue_Pop()`, `barrier_queue_s::list`, `ListTerminate()`, `barrier_queue_s::lock`, `MC_SUCCESS`, `node`, and `RWLOCK_DESTROY`.

Referenced by `mc_platform_Destroy()`.

#### 13.8.1.6 `barrier_node_p barrier_queue_Get (barrier_queue_p list, int id)`

Definition at line 145 of file barrier.c.

References `barrier_queue_s::list`, `list_s::listhead`, `barrier_queue_s::lock`, `listNode_s::next`, `listNode_s::node_data`, `RWLOCK_RDLOCK`, and `RWLOCK_RDUNLOCK`.

Referenced by `MC_Barrier()`, and `MC_BarrierInit()`.

#### 13.8.1.7 `barrier_queue_p barrier_queue_New (void)`

Definition at line 162 of file barrier.c.

References `CHECK_NULL`, `barrier_queue_s::list`, `ListInitialize()`, `barrier_queue_s::lock`, `RWLOCK_INIT`, and `RWLOCK_T`.

Referenced by `mc_platform_Initialize()`.

#### 13.8.1.8 `barrier_node_p barrier_queue_Pop (barrier_queue_p queue)`

Definition at line 176 of file barrier.c.

References `barrier_queue_s::list`, `ListPop()`, and `node`.

Referenced by `barrier_queue_Destroy()`.

## 13.9 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/cmd\_prompt.c File Reference

```
#include <stdio.h>
#include <unistd.h>
#include "config.h"
#include <stdlib.h>
#include <string.h>
#include "include/cmd_prompt.h"
#include "include/commands.h"
```

### Functions

- [cmd\\_prompt\\_p cmd\\_prompt\\_Initialize \(mc\\_platform\\_p mc\\_platform\)](#)
- [int cmd\\_prompt\\_Destroy \(cmd\\_prompt\\_p cmd\\_prompt\)](#)
- [void cmd\\_prompt\\_Start \(mc\\_platform\\_p mc\\_platform\)](#)
- [void \\* cmd\\_prompt\\_Thread \(void \\*arg\)](#)
- [int split\\_string \(char \\*\\*\\*args, const char \\*buf\)](#)
- [int process\\_command \(command\\_t \\*cmd\)](#)
- [int exec\\_command \(command\\_t cmd, mc\\_platform\\_p global\)](#)
- [int dealloc\\_command \(command\\_t \\*cmd\)](#)
- [int handler\\_QUIT \(void \\*arg, mc\\_platform\\_p global\)](#)
- [int handler\\_HELP \(void \\*arg, mc\\_platform\\_p global\)](#)
- [int handler\\_SEND \(void \\*arg, mc\\_platform\\_p global\)](#)
- [int handler\\_PRINT\\_CONNECTLIST \(void \\*arg, mc\\_platform\\_p global\)](#)
- [int handler\\_PRINTLIST\\_MESSAGE \(void \\*arg, mc\\_platform\\_p global\)](#)
- [int handler\\_PRINTLIST\\_AGENTS \(void \\*arg, mc\\_platform\\_p global\)](#)
- [int handler\\_FLUSH\\_AGENTS \(void \\*arg, mc\\_platform\\_p global\)](#)

### 13.9.1 Function Documentation

#### 13.9.1.1 [int cmd\\_prompt\\_Destroy \(cmd\\_prompt\\_p \*cmd\\_prompt\*\)](#)

Definition at line 129 of file cmd\_prompt.c.

References [MC\\_SUCCESS](#).

Referenced by [mc\\_platform\\_Destroy\(\)](#).

#### 13.9.1.2 [cmd\\_prompt\\_p cmd\\_prompt\\_Initialize \(mc\\_platform\\_p \*mc\\_platform\*\)](#)

Definition at line 121 of file cmd\_prompt.c.

Referenced by [mc\\_platform\\_Initialize\(\)](#).

**13.9.1.3 void cmd\_prompt\_Start (mc\_platform\_p mc\_platform)**

Definition at line 136 of file cmd\_prompt.c.

References mc\_platform\_s::cmd\_prompt, cmd\_prompt\_Thread(), MC\_THREAD\_CP, mc\_platform\_s::stack\_size, cmd\_prompt\_s::thread, and THREAD\_CREATE.

Referenced by mc\_platform\_Initialize().

**13.9.1.4 void\* cmd\_prompt\_Thread (void \* arg)**

Definition at line 168 of file cmd\_prompt.c.

References command\_s::args, buf, dealloc\_command(), exec\_command(), command\_s::index, mc\_platform, command\_s::num\_args, process\_command(), and split\_string().

Referenced by cmd\_prompt\_Start().

**13.9.1.5 int dealloc\_command (command\_t \* cmd)**

Definition at line 306 of file cmd\_prompt.c.

References command\_s::args, and command\_s::num\_args.

Referenced by cmd\_prompt\_Thread().

**13.9.1.6 int exec\_command (command\_t cmd, mc\_platform\_p global)**

Definition at line 290 of file cmd\_prompt.c.

References command\_s::args, cmd\_handlers, command\_s::index, and command\_s::num\_args.

Referenced by cmd\_prompt\_Thread().

**13.9.1.7 int handler\_FLUSH\_AGENTS (void \* arg, mc\_platform\_p global)**

Definition at line 401 of file cmd\_prompt.c.

References mc\_platform\_s::agent\_queue, and agent\_queue\_Flush().

**13.9.1.8 int handler\_HELP (void \* arg, mc\_platform\_p global)**

Definition at line 330 of file cmd\_prompt.c.

References command\_s::args, command\_cmds, command\_descriptions, and command\_s::num\_args.

**13.9.1.9 int handler\_PRINT\_CONNECTLIST (void \* arg, mc\_platform\_p global)**

Definition at line 383 of file cmd\_prompt.c.

References mc\_platform\_s::connection\_queue.

**13.9.1.10 int handler\_PRINTLIST\_AGENTS (void \* arg, mc\_platform\_p global)**

Definition at line 395 of file cmd\_prompt.c.

References mc\_platform\_s::agent\_queue.

#### 13.9.1.11 int handler\_PRINTLIST\_MESSAGE (void \* *arg*, mc\_platform\_p *global*)

Definition at line 389 of file cmd\_prompt.c.

References mc\_platform\_s::message\_queue.

#### 13.9.1.12 int handler\_QUIT (void \* *arg*, mc\_platform\_p *global*)

Definition at line 321 of file cmd\_prompt.c.

References COND\_BROADCAST, MUTEX\_LOCK, MUTEX\_UNLOCK, mc\_platform\_s::quit, mc\_platform\_s::quit\_cond, and mc\_platform\_s::quit\_lock.

#### 13.9.1.13 int handler\_SEND (void \* *arg*, mc\_platform\_p *global*)

Definition at line 368 of file cmd\_prompt.c.

References command\_s::args, command\_descriptions, MC\_SendAgentMigrationMessageFile(), and command\_s::num\_args.

#### 13.9.1.14 int process\_command (command\_t \* *cmd*)

Definition at line 272 of file cmd\_prompt.c.

References command\_s::args, command\_cmds, command\_s::index, and command\_s::num\_args.

Referenced by cmd\_prompt\_Thread().

#### 13.9.1.15 int split\_string (char \*\*\* *args*, const char \* *buf*)

Definition at line 224 of file cmd\_prompt.c.

References int.

Referenced by cmd\_prompt\_Thread().

## 13.10 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/connection.c File Reference

```
#include "config.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "include/connection.h"
#include "include/mc_error.h"
#include "include/macros.h"
```

### Functions

- [int connection\\_Destroy](#) ([connection\\_p](#) connection)
- [connection\\_p connection\\_New](#) (void)
- [connection\\_p connection\\_Copy](#) ([connection\\_p](#) connection)

### 13.10.1 Function Documentation

#### 13.10.1.1 [connection\\_p connection\\_Copy](#) ([connection\\_p](#) *connection*)

Definition at line 80 of file connection.c.

References [connection\\_s::addr](#), [connection\\_s::clientfd](#), [connection\\_s::connect\\_id](#), [connection\\_New\(\)](#), [connection\\_s::remote\\_hostname](#), and [connection\\_s::serverfd](#).

#### 13.10.1.2 [int connection\\_Destroy](#) ([connection\\_p](#) *connection*)

Definition at line 48 of file connection.c.

References [connection\\_s::clientfd](#), [MC\\_SUCCESS](#), and [connection\\_s::remote\\_hostname](#).

Referenced by [acc\\_connection\\_Thread\(\)](#).

#### 13.10.1.3 [connection\\_p connection\\_New](#) (void)

Definition at line 69 of file connection.c.

Referenced by [connection\\_Copy\(\)](#), and [listen\\_Thread\(\)](#).

## 13.11 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/data\_structures.c File Reference

```
#include "config.h"
#include "include/ap_queue_template.h"
#include "include/data_structures.h"
#include "include/interpreter_variable_data.h"
#include "include/agent.h"
```

### Functions

- [AP\\_QUEUE\\_SEARCH\\_TEMPLATE](#) (AP\_QUEUE\_REMOVE\_TEMPLATE(connection\_queue, [Search](#), AP\_QUEUE\_REMOVE\_TEMPLATE(connection, AP\_QUEUE\_REMOVE\_TEMPLATE(int,([node](#)->connect\_id==key)))
- [AP\\_QUEUE\\_SEARCH\\_TEMPLATE](#) (AP\_QUEUE\_REMOVE\_TEMPLATE(agent\_variable\_list, AP\_QUEUE\_REMOVE\_TEMPLATE([Search](#), AP\_QUEUE\_REMOVE\_TEMPLATE(interpreter\_variable\_data, char \*,(!strcmp([node](#)->name, key))))
- [AP\\_QUEUE\\_STD\\_DEFN\\_TEMPLATE](#) (AP\_QUEUE\_SEARCH\_TEMPLATE(agent\_queue, AP\_QUEUE\_SEARCH\_TEMPLATE(agent)
- [int agent\\_queue\\_Flush](#) (agent\_queue\_p queue)
- [AP\\_QUEUE\\_STD\\_DEFN\\_TEMPLATE](#) (fipa\_acl\_message\_p mail\_queue\_SearchReceivers(mail\_queue\_p mail\_queue, fipa\_acl\_message)
- [AP\\_QUEUE\\_STD\\_DEFN\\_TEMPLATE](#) (mailbox\_queue, agent\_mailbox)
- [int AP\\_GENERIC\\_Destroy](#) (AP\_GENERIC\_t \*blah)
- [AP\\_GENERIC\\_p AP\\_GENERIC\\_Copy](#) (AP\_GENERIC\_p AP\_GENERIC)

### 13.11.1 Function Documentation

#### 13.11.1.1 int agent\_queue\_Flush (agent\_queue\_p queue)

Definition at line 197 of file data\_structures.c.

References [agent\\_s::agent\\_status](#), [agent\\_s::id](#), [ListSearch\(\)](#), [MUTEX\\_LOCK](#), [MUTEX\\_UNLOCK](#), [agent\\_s::name](#), and [node](#).

Referenced by [handler\\_FLUSH\\_AGENTS\(\)](#).

#### 13.11.1.2 AP\_GENERIC\_p AP\_GENERIC\_Copy (AP\_GENERIC\_p AP\_GENERIC)

Definition at line 292 of file data\_structures.c.

#### 13.11.1.3 int AP\_GENERIC\_Destroy (AP\_GENERIC\_t \* blah)

Definition at line 288 of file data\_structures.c.

**13.11.1.4** `AP_QUEUE_SEARCH_TEMPLATE (AP_QUEUE_REMOVE_TEMPLATE(  
agent_variable_list, AP_QUEUE_REMOVE_TEMPLATE( Search,  
AP_QUEUE_REMOVE_TEMPLATE( interpreter_variable_data, char *,  
(!strcmp(node->name, key)))`

Definition at line 84 of file data\_structures.c.

References message\_s::from\_address, ListSearch(), message\_s::message\_id, MUTEX\_LOCK, MUTEX\_UNLOCK, node, and message\_s::to\_address.

**13.11.1.5** `AP_QUEUE_SEARCH_TEMPLATE (AP_QUEUE_REMOVE_TEMPLATE(  
connection_queue, Search, AP_QUEUE_REMOVE_TEMPLATE( connection,  
AP_QUEUE_REMOVE_TEMPLATE( int, (node->connect_id==key))`

Definition at line 45 of file data\_structures.c.

References connection\_s::connect\_id, ListSearch(), MUTEX\_LOCK, MUTEX\_UNLOCK, node, and connection\_s::remote\_hostname.

**13.11.1.6** `AP_QUEUE_STD_DEFN_TEMPLATE (mailbox_queue, agent_mailbox)`

Definition at line 253 of file data\_structures.c.

References agent\_ChScriptInitVar(), COND\_SIGNAL, ListPop(), MUTEX\_LOCK, and MUTEX\_UNLOCK.

**13.11.1.7** `AP_QUEUE_STD_DEFN_TEMPLATE (fipa_acl_message_p  
mail_queue_SearchReceivers( mail_queue_p mail_queue, fipa_acl_message)`

Definition at line 214 of file data\_structures.c.

References fipa\_agent\_identifier\_set\_s::fipa\_agent\_identifiers, MUTEX\_LOCK, MUTEX\_UNLOCK, fipa\_agent\_identifier\_s::name, listNode\_s::next, node, listNode\_s::node\_data, and fipa\_acl\_message\_s::receiver.

**13.11.1.8** `AP_QUEUE_STD_DEFN_TEMPLATE (AP_QUEUE_SEARCH_TEMPLATE(  
agent_queue, AP_QUEUE_SEARCH_TEMPLATE( agent)`

Definition at line 145 of file data\_structures.c.

References agent\_s::agent\_status, agent\_s::id, ListSearch(), MUTEX\_LOCK, MUTEX\_UNLOCK, agent\_s::name, and node.



## 13.12 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/df.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include "config.h"
#include "include/mc_platform.h"
#include "include/df.h"
#include "include/df_request.x.h"
```

### Defines

- #define [REQUEST](#)(name, string, description)

### Functions

- [int df\\_Add](#) (struct df\_s \*df, struct df\_node\_s \*[node](#))
- [int df\\_AddRequest](#) (struct df\_s \*df, struct df\_request\_list\_node\_s \*[node](#))
- [int df\\_Destroy](#) (df\_p df)
- df\_p [df\\_Initialize](#) (mc\_platform\_p mc\_platform)
- [int df\\_ProcessRequest](#) (struct mc\_platform\_s \*global)
- [int df\\_SearchForService](#) (df\_p df, const char \*searchstring, char \*\*\*agent\_names, char \*\*\*service\_names, int \*\*agent\_ids, int \*num\_entries)
- void [df\\_Start](#) (mc\_platform\_p mc\_platform)
- [int df\\_request\\_list\\_node\\_Destroy](#) (df\_request\_list\_node\_p [node](#))
- df\_request\_list\_node\_p [df\\_request\\_list\\_node\\_New](#) (void)
- [int df\\_request\\_list\\_Destroy](#) (df\_request\_list\_p df\_request\_list)
- df\_request\_list\_p [df\\_request\\_list\\_New](#) (void)
- df\_request\_list\_node\_p [df\\_request\\_list\\_Pop](#) (df\_request\_list\_p requests)
- df\_request\_search\_p [df\\_request\\_search\\_New](#) (void)
- [int df\\_request\\_search\\_Destroy](#) (df\_request\_search\_p [node](#))
- [int df\\_node\\_Destroy](#) (df\_node\_p df\_node)
- void \* [df\\_Thread](#) (void \*arg)
- [int request\\_handler\\_REGISTER](#) (struct mc\_platform\_s \*global, void \*data)
- [int request\\_handler\\_SEARCH](#) (struct mc\_platform\_s \*global, void \*data)
- [int request\\_handler\\_SUBSCRIBE](#) (struct mc\_platform\_s \*global, void \*data)
- [int request\\_handler\\_DEREGISTER](#) (struct mc\_platform\_s \*global, void \*data)

### 13.12.1 Define Documentation

#### 13.12.1.1 #define REQUEST(name, string, description)

Value:

```

if ( !strcmp(request->command, string ) ) { \
    return_code = MC_SUCCESS; \
    handler_code = request_handler_##name( \
        global, \
        request->data ); \
    request_code = REQUEST_##name; \
} else

```

Referenced by `acc_connection_Thread()`, and `acc_MessageHandlerThread()`.

## 13.12.2 Function Documentation

### 13.12.2.1 `int df_Add (struct df_s * df, struct df_node_s * node)`

Definition at line 50 of file `df.c`.

References `ListAdd()`, `MC_SUCCESS`, and `SIGNAL`.

Referenced by `request_handler_REGISTER()`.

### 13.12.2.2 `int df_AddRequest (struct df_s * df, struct df_request_list_node_s * node)`

Definition at line 66 of file `df.c`.

References `ListAdd()`, and `SIGNAL`.

Referenced by `MC_DeregisterService()`, `MC_RegisterService()`, and `MC_SearchForService()`.

### 13.12.2.3 `int df_Destroy (df_p df)`

Definition at line 83 of file `df.c`.

References `COND_DESTROY`, `df_node_Destroy()`, `df_request_list_Destroy()`, `ListPop()`, `ListTerminate()`, `MC_SUCCESS`, `MUTEX_DESTROY`, and `MUTEX_LOCK`.

Referenced by `mc_platform_Destroy()`.

### 13.12.2.4 `df_p df_Initialize (mc_platform_p mc_platform)`

Definition at line 101 of file `df.c`.

References `COND_INIT`, `COND_T`, `df_request_list_New()`, `ListInitialize()`, `MUTEX_INIT`, and `MUTEX_T`.

Referenced by `mc_platform_Initialize()`.

### 13.12.2.5 `int df_node_Destroy (df_node_p df_node)`

Definition at line 412 of file `df.c`.

References `MC_SUCCESS`, and `MUTEX_LOCK`.

Referenced by `df_Destroy()`.

**13.12.2.6 int df\_ProcessRequest (struct mc\_platform\_s \* *global*)**

Definition at line 132 of file df.c.

References mc\_platform\_s::df, df\_request\_list\_Pop(), MC\_ERR\_EMPTY, and MC\_ERR\_INVALID.

Referenced by df\_Thread().

**13.12.2.7 int df\_request\_list\_Destroy (df\_request\_list\_p *df\_request\_list*)**

Definition at line 321 of file df.c.

References df\_request\_list\_node\_Destroy(), ListPop(), ListTerminate(), MC\_SUCCESS, and node.

Referenced by df\_Destroy().

**13.12.2.8 df\_request\_list\_p df\_request\_list\_New (void)**

Definition at line 343 of file df.c.

References CHECK\_NULL, COND\_INIT, COND\_T, ListInitialize(), MUTEX\_INIT, and MUTEX\_T.

Referenced by df\_Initialize().

**13.12.2.9 int df\_request\_list\_node\_Destroy (df\_request\_list\_node\_p *node*)**

Definition at line 292 of file df.c.

References COND\_DESTROY, MC\_SUCCESS, and MUTEX\_DESTROY.

Referenced by df\_request\_list\_Destroy(), and MC\_SearchForService().

**13.12.2.10 df\_request\_list\_node\_p df\_request\_list\_node\_New (void)**

Definition at line 303 of file df.c.

References CHECK\_NULL, COND\_INIT, COND\_T, MUTEX\_INIT, MUTEX\_T, and node.

Referenced by MC\_DeregisterService(), MC\_RegisterService(), and MC\_SearchForService().

**13.12.2.11 df\_request\_list\_node\_p df\_request\_list\_Pop (df\_request\_list\_p *requests*)**

Definition at line 368 of file df.c.

References ListPop(), MUTEX\_LOCK, MUTEX\_UNLOCK, and node.

Referenced by df\_ProcessRequest().

**13.12.2.12 int df\_request\_search\_Destroy (df\_request\_search\_p *node*)**

Definition at line 399 of file df.c.

References COND\_DESTROY, MC\_SUCCESS, and MUTEX\_DESTROY.

Referenced by MC\_SearchForService().

**13.12.2.13 df\_request\_search\_p df\_request\_search\_New (void)**

Definition at line 384 of file df.c.

References CHECK\_NULL, COND\_INIT, COND\_T, MUTEX\_INIT, MUTEX\_T, and search.

Referenced by MC\_SearchForService().

**13.12.2.14 int df\_SearchForService (df\_p df, const char \* searchstring, char \*\*\* agent\_names, char \*\*\* service\_names, int \*\* agent\_ids, int \* num\_entries)**

Definition at line 176 of file df.c.

References MC\_ERR\_NOT\_FOUND, MC\_SUCCESS, MUTEX\_LOCK, MUTEX\_UNLOCK, listNode\_s::next, and listNode\_s::node\_data.

Referenced by request\_handler\_SEARCH().

**13.12.2.15 void df\_Start (mc\_platform\_p mc\_platform)**

Definition at line 261 of file df.c.

References mc\_platform\_s::df, df\_Thread(), MC\_THREAD\_DF, mc\_platform\_s::stack\_size, and THREAD\_CREATE.

Referenced by mc\_platform\_Initialize().

**13.12.2.16 void\* df\_Thread (void \* arg)**

Definition at line 426 of file df.c.

References COND\_BROADCAST, COND\_WAIT, mc\_platform\_s::df, df\_ProcessRequest(), MC\_SUCCESS, MUTEX\_LOCK, MUTEX\_UNLOCK, mc\_platform\_s::quit, mc\_platform\_s::quit\_lock, and THREAD\_EXIT.

Referenced by df\_Start().

**13.12.2.17 int request\_handler\_DEREGISTER (struct mc\_platform\_s \* global, void \* data)**

Definition at line 520 of file df.c.

References mc\_platform\_s::df, MC\_SUCCESS, MUTEX\_LOCK, MUTEX\_UNLOCK, listNode\_s::next, node, and listNode\_s::node\_data.

**13.12.2.18 int request\_handler\_REGISTER (struct mc\_platform\_s \* global, void \* data)**

Definition at line 490 of file df.c.

References mc\_platform\_s::df, and df\_Add().

**13.12.2.19 int request\_handler\_SEARCH (struct mc\_platform\_s \* global, void \* data)**

Definition at line 496 of file df.c.

References mc\_platform\_s::df, df\_SearchForService(), MC\_SUCCESS, search, and SIGNAL.

**13.12.2.20** `int request_handler_SUBSCRIBE (struct mc_platform_s * global, void * data)`

Definition at line 515 of file df.c.

## 13.13 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/dynstring.c File Reference

```
#include "config.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "include/dynstring.h"
```

### Functions

- [dynstring\\_t \\* dynstring\\_New](#) (void)
- [int dynstring\\_Append](#) ([dynstring\\_t](#) \*msg, char \*str)
- [int dynstring\\_Destroy](#) ([dynstring\\_t](#) \*dynstring)

### 13.13.1 Function Documentation

#### 13.13.1.1 [int dynstring\\_Append](#) ([dynstring\\_t](#) \* msg, char \* str)

Definition at line 30 of file `dynstring.c`.

References `COMPOSE_BLOCKSIZE`, `dynstring_s::len`, `dynstring_s::message`, and `dynstring_s::size`.

Referenced by `fipa_acl_Compose()`, `fipa_agent_identifier_Compose()`, `fipa_agent_identifier_set_Compose()`, `fipa_DateTime_Compose()`, `fipa_envelope_Compose__from()`, `fipa_number_Compose()`, `fipa_performative_Compose()`, `fipa_protocol_Compose()`, `fipa_string_Compose()`, `fipa_url_Compose()`, `fipa_url_sequence_Compose()`, `fipa_word_Compose()`, `message_send_Thread()`, `mtp_http_CreateMessage()`, and `mtp_http_InitializeFromConnection()`.

#### 13.13.1.2 [int dynstring\\_Destroy](#) ([dynstring\\_t](#) \* *dynstring*)

Definition at line 56 of file `dynstring.c`.

References `dynstring_s::message`.

Referenced by `fipa_envelope_Compose__from()`, `MC_AclSend()`, `message_send_Thread()`, `mtp_http_CreateMessage()`, and `mtp_http_InitializeFromConnection()`.

#### 13.13.1.3 [dynstring\\_t\\* dynstring\\_New](#) (void)

Definition at line 14 of file `dynstring.c`.

References `COMPOSE_BLOCKSIZE`, `dynstring_s::len`, `dynstring_s::message`, and `dynstring_s::size`.

Referenced by `fipa_acl_Compose()`, `fipa_envelope_Compose__from()`, `message_send_Thread()`, `mtp_http_CreateMessage()`, and `mtp_http_InitializeFromConnection()`.

## 13.14 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/fipa\_acl.c File Reference

```
#include "config.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <strings.h>
#include "include/fipa_acl.h"
#include "include/mc_error.h"
#include "include/macros.h"
```

### Defines

- #define [FREEMEM](#)(x) if (x != NULL) free(x)

### Functions

- [fipa\\_acl\\_message\\_t \\* fipa\\_acl\\_message\\_New](#) (void)
- [int fipa\\_acl\\_message\\_Destroy](#) (fipa\_acl\_message\_t \*message)
- [fipa\\_acl\\_message\\_t \\* fipa\\_acl\\_message\\_Copy](#) (fipa\_acl\_message\_t \*src)
- [fipa\\_message\\_string\\_t \\* fipa\\_message\\_string\\_New](#) (void)
- [int fipa\\_message\\_string\\_Destroy](#) (fipa\_message\_string\_t \*message)
- [fipa\\_message\\_string\\_t \\* fipa\\_message\\_string\\_Copy](#) (fipa\_message\_string\_t \*src)
- [fipa\\_url\\_sequence\\_t \\* fipa\\_url\\_sequence\\_New](#) (void)
- [int fipa\\_url\\_sequence\\_Destroy](#) (fipa\_url\_sequence\_t \*sequence)
- [fipa\\_url\\_sequence\\_t \\* fipa\\_url\\_sequence\\_Copy](#) (fipa\_url\_sequence\_t \*src)
- [fipa\\_agent\\_identifier\\_set\\_t \\* fipa\\_agent\\_identifier\\_set\\_New](#) (void)
- [int fipa\\_agent\\_identifier\\_set\\_Destroy](#) (fipa\_agent\_identifier\_set\_t \*idset)
- [fipa\\_agent\\_identifier\\_set\\_t \\* fipa\\_agent\\_identifier\\_set\\_Copy](#) (fipa\_agent\_identifier\_set\_t \*src)
- [fipa\\_agent\\_identifier\\_t \\* fipa\\_agent\\_identifier\\_New](#) (void)
- [int fipa\\_agent\\_identifier\\_Destroy](#) (fipa\_agent\_identifier\_t \*id)
- [fipa\\_agent\\_identifier\\_t \\* fipa\\_agent\\_identifier\\_Copy](#) (fipa\_agent\_identifier\_t \*src)
- [fipa\\_expression\\_t \\* fipa\\_expression\\_New](#) (void)
- [int fipa\\_expression\\_Destroy](#) (fipa\_expression\_t \*expr)
- [fipa\\_expression\\_t \\* fipa\\_expression\\_Copy](#) (fipa\_expression\_t \*src)
- [fipa\\_word\\_t \\* fipa\\_word\\_New](#) (void)
- [int fipa\\_word\\_Destroy](#) (fipa\_word\_t \*word)
- [fipa\\_word\\_t \\* fipa\\_word\\_Copy](#) (fipa\_word\_t \*src)
- [fipa\\_string\\_t \\* fipa\\_string\\_New](#) (void)
- [int fipa\\_string\\_Destroy](#) (fipa\_string\_t \*str)
- [fipa\\_string\\_t \\* fipa\\_string\\_Copy](#) (fipa\_string\_t \*src)
- [fipa\\_DateTime\\_t \\* fipa\\_DateTime\\_New](#) (void)
- [int fipa\\_DateTime\\_Destroy](#) (fipa\_DateTime\_t \*dt)
- [fipa\\_DateTime\\_t \\* fipa\\_DateTime\\_Copy](#) (fipa\_DateTime\_t \*src)
- [fipa\\_url\\_t \\* fipa\\_url\\_New](#) (void)

- `int fipa_url_Destroy (fipa_url_t *url)`
- `fipa_url_t * fipa_url_Copy (fipa_url_t *src)`
- `fipa_number_t * fipa_number_New (void)`
- `int fipa_number_Destroy (fipa_number_t *number)`
- `fipa_number_t * fipa_number_Copy (fipa_number_t *src)`
- `int fipa_acl_Parse (fipa_acl_message_p acl, fipa_message_string_p message)`
- `int fipa_message_parameter_Parse (fipa_acl_message_p acl, fipa_message_string_p message)`
- `int fipa_protocol_type_Parse (enum fipa_protocol_e *protocol, fipa_message_string_p message)`
- `int fipa_message_type_Parse (enum fipa_performative_e *performative, fipa_message_string_p message)`
- `int fipa_GetAtom (fipa_message_string_p message, char expected_atom)`
- `int fipa_word_Parse (fipa_word_t **word, fipa_message_string_p message)`
- `int fipa_CheckNextToken (const fipa_message_string_p message, const char *token)`
- `int fipa_expression_Parse (fipa_expression_t **expression, fipa_message_string_p message)`
- `int fipa_GetNextWord (char **word, const fipa_message_string_p message)`
- `int fipa_GetWholeToken (char **word, fipa_message_string_p message)`
- `int fipa_datetime_Parse (fipa_DateTime_p *datetime, fipa_message_string_p message)`
- `int fipa_string_Parse (fipa_string_p *fipa_string, fipa_message_string_p message)`
- `int fipa_agent_identifier_Parse (fipa_agent_identifier_p *aid, fipa_message_string_p message)`
- `int fipa_url_sequence_Parse (fipa_url_sequence_p *urls, fipa_message_string_p message)`
- `int fipa_url_Parse (fipa_url_p *url, fipa_message_string_p message)`
- `int fipa_agent_identifier_set_Parse (fipa_agent_identifier_set_p *agent_ids, fipa_message_string_p message)`
- `int fipa_acl_Compose (dynstring_t **msg, fipa_acl_message_t *acl)`
- `int fipa_protocol_Compose (dynstring_t *msg, enum fipa_protocol_e protocol)`
- `int fipa_performative_Compose (dynstring_t *msg, enum fipa_performative_e performative)`
- `int fipa_url_sequence_Compose (dynstring_t *msg, fipa_url_sequence_t *urls)`
- `int fipa_agent_identifier_set_Compose (dynstring_t *msg, fipa_agent_identifier_set_t *ids)`
- `int fipa_agent_identifier_Compose (dynstring_t *msg, fipa_agent_identifier_t *id)`
- `int fipa_expression_Compose (dynstring_t *msg, fipa_expression_t *expr)`
- `int fipa_word_Compose (dynstring_t *msg, fipa_word_t *word)`
- `int fipa_string_Compose (dynstring_t *msg, fipa_string_t *string)`
- `int fipa_DateTime_Compose (dynstring_t *msg, fipa_DateTime_t *date)`
- `int fipa_url_Compose (dynstring_t *msg, fipa_url_t *url)`
- `int fipa_number_Compose (dynstring_t *msg, fipa_number_t *number)`
- `struct fipa_acl_message_s * fipa_Reply (struct fipa_acl_message_s *acl)`

### 13.14.1 Define Documentation

#### 13.14.1.1 #define FREEMEM(x) if (x != NULL) free(x)

Definition at line 53 of file `fipa_acl.c`.

Referenced by `fipa_expression_Destroy()`.



## 13.14.2 Function Documentation

### 13.14.2.1 `int fipa_acl_Compose (dynstring_t **msg, fipa_acl_message_t *acl)`

Definition at line 1232 of file fipa\_acl.c.

References `fipa_acl_message_s::content`, `fipa_acl_message_s::conversation_id`, `dynstring_Append()`, `dynstring_New()`, `fipa_acl_message_s::encoding`, `fipa_agent_identifier_Compose()`, `fipa_agent_identifier_set_Compose()`, `fipa_DateTime_Compose()`, `fipa_expression_Compose()`, `fipa_performative_Compose()`, `fipa_protocol_Compose()`, `FIPA_PROTOCOL_NONE`, `fipa_string_Compose()`, `fipa_acl_message_s::in_reply_to`, `fipa_acl_message_s::language`, `fipa_acl_message_s::ontology`, `fipa_acl_message_s::performative`, `fipa_acl_message_s::protocol`, `fipa_acl_message_s::receiver`, `fipa_acl_message_s::reply_by`, `fipa_acl_message_s::reply_to`, `fipa_acl_message_s::reply_with`, and `fipa_acl_message_s::sender`.

Referenced by `MC_AclSend()`.

### 13.14.2.2 `fipa_acl_message_t* fipa_acl_message_Copy (fipa_acl_message_t *src)`

Definition at line 86 of file fipa\_acl.c.

References `fipa_acl_message_s::content`, `fipa_acl_message_s::conversation_id`, `fipa_acl_message_s::encoding`, `fipa_acl_message_New()`, `fipa_agent_identifier_Copy()`, `fipa_agent_identifier_set_Copy()`, `fipa_DateTime_Copy()`, `fipa_expression_Copy()`, `fipa_string_Copy()`, `fipa_acl_message_s::in_reply_to`, `fipa_acl_message_s::language`, `fipa_acl_message_s::ontology`, `fipa_acl_message_s::performative`, `fipa_acl_message_s::protocol`, `fipa_acl_message_s::receiver`, `fipa_acl_message_s::reply_by`, `fipa_acl_message_s::reply_to`, `fipa_acl_message_s::reply_with`, and `fipa_acl_message_s::sender`.

Referenced by `MC_AclSend()`.

### 13.14.2.3 `int fipa_acl_message_Destroy (fipa_acl_message_t *message)`

Definition at line 65 of file fipa\_acl.c.

References `fipa_acl_message_s::content`, `fipa_acl_message_s::conversation_id`, `fipa_acl_message_s::encoding`, `fipa_agent_identifier_Destroy()`, `fipa_agent_identifier_set_Destroy()`, `fipa_DateTime_Destroy()`, `fipa_expression_Destroy()`, `fipa_string_Destroy()`, `fipa_acl_message_s::in_reply_to`, `fipa_acl_message_s::language`, `fipa_acl_message_s::ontology`, `fipa_acl_message_s::receiver`, `fipa_acl_message_s::reply_by`, `fipa_acl_message_s::reply_to`, `fipa_acl_message_s::reply_with`, and `fipa_acl_message_s::sender`.

Referenced by `acc_connection_Thread()`, and `MC_AclDestroy()`.

### 13.14.2.4 `fipa_acl_message_t* fipa_acl_message_New (void)`

Definition at line 57 of file fipa\_acl.c.

Referenced by `acc_connection_Thread()`, `fipa_acl_message_Copy()`, `fipa_Reply()`, and `MC_AclNew()`.

### 13.14.2.5 `int fipa_acl_Parse (fipa_acl_message_p acl, fipa_message_string_p message)`

Definition at line 464 of file fipa\_acl.c.

References `fipa_GetAtom()`, `fipa_message_parameter_Parse()`, `fipa_message_type_Parse()`, `MC_ERR_PARSE`, and `fipa_acl_message_s::performative`.

Referenced by `acc_connection_Thread()`.

#### 13.14.2.6 `int fipa_agent_identifier_Compose (dynstring_t * msg, fipa_agent_identifier_t * id)`

Definition at line 1446 of file `fipa_acl.c`.

References `fipa_agent_identifier_s::addresses`, `dynstring_Append()`, `fipa_agent_identifier_set_Compose()`, `fipa_url_sequence_Compose()`, `fipa_agent_identifier_s::name`, `fipa_agent_identifier_set_s::num`, `fipa_url_sequence_s::num`, and `fipa_agent_identifier_s::resolvers`.

Referenced by `fipa_acl_Compose()`, and `fipa_agent_identifier_set_Compose()`.

#### 13.14.2.7 `fipa_agent_identifier_t* fipa_agent_identifier_Copy (fipa_agent_identifier_t * src)`

Definition at line 235 of file `fipa_acl.c`.

References `fipa_agent_identifier_s::addresses`, `fipa_agent_identifier_New()`, `fipa_agent_identifier_set_Copy()`, `fipa_url_sequence_Copy()`, `fipa_agent_identifier_s::name`, and `fipa_agent_identifier_s::resolvers`.

Referenced by `fipa_acl_message_Copy()`, `fipa_acl_Param_Copy()`, `fipa_agent_identifier_set_Copy()`, and `fipa_Reply()`.

#### 13.14.2.8 `int fipa_agent_identifier_Destroy (fipa_agent_identifier_t * id)`

Definition at line 223 of file `fipa_acl.c`.

References `fipa_agent_identifier_s::addresses`, `fipa_agent_identifier_set_Destroy()`, `fipa_url_sequence_Destroy()`, `fipa_agent_identifier_s::name`, and `fipa_agent_identifier_s::resolvers`.

Referenced by `fipa_acl_message_Destroy()`, `fipa_acl_Param_Destroy()`, `fipa_agent_identifier_set_Destroy()`, and `MC_AclSetSender()`.

#### 13.14.2.9 `fipa_agent_identifier_t* fipa_agent_identifier_New (void)`

Definition at line 215 of file `fipa_acl.c`.

Referenced by `fipa_agent_identifier_Copy()`, `fipa_envelope_ParseAgentIdentifier()`, `MC_AclAddReceiver()`, `MC_AclAddReplyTo()`, and `MC_AclSetSender()`.

#### 13.14.2.10 `int fipa_agent_identifier_Parse (fipa_agent_identifier_p * aid, fipa_message_string_p message)`

Definition at line 1032 of file `fipa_acl.c`.

References `CHECK_NULL`, `fipa_word_s::content`, `fipa_agent_identifier_set_Parse()`, `fipa_GetAtom()`, `fipa_url_sequence_Parse()`, `fipa_word_Destroy()`, `fipa_word_Parse()`, `MC_ERR_PARSE`, `MC_SUCCESS`, and `fipa_message_string_s::parse`.

Referenced by `fipa_agent_identifier_set_Parse()`, and `fipa_message_parameter_Parse()`.

#### 13.14.2.11 `int fipa_agent_identifier_set_Compose (dynstring_t * msg, fipa_agent_identifier_set_t * ids)`

Definition at line 1433 of file `fipa_acl.c`.

References `dynstring_Append()`, `fipa_agent_identifier_Compose()`, `fipa_agent_identifier_set_s::fipa_agent_identifiers`, and `fipa_agent_identifier_set_s::num`.

Referenced by `fipa_acl_Compose()`, and `fipa_agent_identifier_Compose()`.

#### 13.14.2.12 `fipa_agent_identifier_set_t* fipa_agent_identifier_set_Copy (fipa_agent_identifier_set_t * src)`

Definition at line 194 of file `fipa_acl.c`.

References `fipa_agent_identifier_Copy()`, `fipa_agent_identifier_set_New()`, `fipa_agent_identifier_set_s::fipa_agent_identifiers`, `fipa_agent_identifier_set_s::num`, and `fipa_agent_identifier_set_s::retain_order`.

Referenced by `fipa_acl_message_Copy()`, `fipa_acl_Param_Copy()`, `fipa_agent_identifier_Copy()`, and `fipa_Reply()`.

#### 13.14.2.13 `int fipa_agent_identifier_set_Destroy (fipa_agent_identifier_set_t * idset)`

Definition at line 182 of file `fipa_acl.c`.

References `fipa_agent_identifier_Destroy()`, `fipa_agent_identifier_set_s::fipa_agent_identifiers`, and `fipa_agent_identifier_set_s::num`.

Referenced by `fipa_acl_message_Destroy()`, `fipa_acl_Param_Destroy()`, and `fipa_agent_identifier_Destroy()`.

#### 13.14.2.14 `fipa_agent_identifier_set_t* fipa_agent_identifier_set_New (void)`

Definition at line 174 of file `fipa_acl.c`.

Referenced by `fipa_agent_identifier_set_Copy()`, `fipa_envelope_HandleIntendedReceiver()`, `fipa_envelope_HandleTo()`, `fipa_Reply()`, `MC_AclAddReceiver()`, and `MC_AclAddReplyTo()`.

#### 13.14.2.15 `int fipa_agent_identifier_set_Parse (fipa_agent_identifier_set_p * agent_ids, fipa_message_string_p message)`

Definition at line 1191 of file `fipa_acl.c`.

References `fipa_word_s::content`, `fipa_agent_identifier_Parse()`, `fipa_GetAtom()`, `fipa_word_Parse()`, `MC_ERR_PARSE`, `MC_SUCCESS`, and `fipa_agent_identifier_set_s::retain_order`.

Referenced by `fipa_agent_identifier_Parse()`, and `fipa_message_parameter_Parse()`.

#### 13.14.2.16 `int fipa_CheckNextToken (const fipa_message_string_p message, const char * token)`

Definition at line 694 of file `fipa_acl.c`.

References `fipa_message_string_s::parse`.

Referenced by `fipa_expression_Parse()`.

#### 13.14.2.17 `int fipa_DateTime_Compose (dynstring_t * msg, fipa_DateTime_t * date)`

Definition at line 1519 of file `fipa_acl.c`.

References `buf`, `fipa_DateTime_s::day`, `dynstring_Append()`, `fipa_DateTime_s::hour`, `fipa_DateTime_s::millisecond`, `fipa_DateTime_s::minute`, `fipa_DateTime_s::month`, `fipa_DateTime_s::second`, `fipa_DateTime_s::sign`, and `fipa_DateTime_s::year`.

Referenced by `fipa_acl_Compose()`, and `fipa_expression_Compose()`.

#### 13.14.2.18 `fipa_DateTime_t* fipa_DateTime_Copy (fipa_DateTime_t * src)`

Definition at line 398 of file `fipa_acl.c`.

References `fipa_DateTime_New()`.

Referenced by `fipa_acl_envelope_Received_Copy()`, `fipa_acl_message_Copy()`, `fipa_acl_Param_Copy()`, and `fipa_expression_Copy()`.

#### 13.14.2.19 `int fipa_DateTime_Destroy (fipa_DateTime_t * dt)`

Definition at line 391 of file `fipa_acl.c`.

Referenced by `fipa_acl_envelope_Received_Destroy()`, `fipa_acl_message_Destroy()`, `fipa_acl_Param_Destroy()`, and `fipa_expression_Destroy()`.

#### 13.14.2.20 `fipa_DateTime_t* fipa_DateTime_New (void)`

Definition at line 383 of file `fipa_acl.c`.

Referenced by `fipa_DateTime_Copy()`.

#### 13.14.2.21 `int fipa_datetime_Parse (fipa_DateTime_p * datetime, fipa_message_string_p message)`

Definition at line 849 of file `fipa_acl.c`.

References `buf`, `fipa_GetWholeToken()`, `MC_ERR_PARSE`, `MC_SUCCESS`, `fipa_message_string_s::parse`, and `fipa_DateTime_s::sign`.

Referenced by `fipa_envelope_HandleDate()`, `fipa_envelope_HandleReceived()`, `fipa_expression_Parse()`, and `fipa_message_parameter_Parse()`.

#### 13.14.2.22 `int fipa_expression_Compose (dynstring_t * msg, fipa_expression_t * expr)`

Definition at line 1472 of file `fipa_acl.c`.

References `fipa_expression_s::content`, `fipa_expression_s::content_u::datetime`, `fipa_expression_s::content_u::expression`, `fipa_DateTime_Compose()`, `FIPA_EXPR_DATETIME`, `FIPA_EXPR_EXPRESSION`, `FIPA_EXPR_NUMBER`, `FIPA_EXPR_STRING`, `FIPA_EXPR_WORD`, `fipa_expression_Compose()`, `fipa_number_Compose()`, `fipa_string_Compose()`, `fipa_word_Compose()`, `MC_ERR_PARSE`, `fipa_expression_s::content_u::number`, `fipa_expression_s::content_u::string`, `fipa_expression_s::type`, and `fipa_expression_s::content_u::word`.

Referenced by `fipa_acl_Compose()`, and `fipa_expression_Compose()`.

#### 13.14.2.23 `fipa_expression_t* fipa_expression_Copy (fipa_expression_t * src)`

Definition at line 286 of file `fipa_acl.c`.

References `fipa_expression_s::content`, `fipa_expression_s::content_u::datetime`, `fipa_expression_s::content_u::expression`, `fipa_DateTime_Copy()`, `FIPA_EXPR_DATETIME`, `FIPA_EXPR_EXPRESSION`, `FIPA_EXPR_NUMBER`, `FIPA_EXPR_STRING`, `FIPA_EXPR_WORD`, `fipa_expression_Copy()`, `fipa_expression_Destroy()`, `fipa_expression_New()`, `fipa_number_Copy()`, `fipa_string_Copy()`, `fipa_word_Copy()`, `fipa_expression_s::content_u::number`, `fipa_expression_s::content_u::string`, `fipa_expression_s::type`, and `fipa_expression_s::content_u::word`.

Referenced by `fipa_acl_message_Copy()`, `fipa_expression_Copy()`, and `fipa_Reply()`.

#### 13.14.2.24 `int fipa_expression_Destroy (fipa_expression_t * expr)`

Definition at line 255 of file `fipa_acl.c`.

References `fipa_expression_s::content`, `fipa_expression_s::content_u::datetime`, `fipa_expression_s::content_u::expression`, `fipa_DateTime_Destroy()`, `FIPA_EXPR_DATETIME`, `FIPA_EXPR_EXPRESSION`, `FIPA_EXPR_NUMBER`, `FIPA_EXPR_STRING`, `FIPA_EXPR_WORD`, `fipa_expression_Destroy()`, `fipa_number_Destroy()`, `fipa_string_Destroy()`, `fipa_word_Destroy()`, `FREEMEM`, `fipa_expression_s::content_u::number`, `fipa_expression_s::content_u::string`, `fipa_expression_s::type`, and `fipa_expression_s::content_u::word`.

Referenced by `fipa_acl_message_Destroy()`, `fipa_expression_Copy()`, and `fipa_expression_Destroy()`.

#### 13.14.2.25 `fipa_expression_t* fipa_expression_New (void)`

Definition at line 247 of file `fipa_acl.c`.

Referenced by `fipa_expression_Copy()`, and `MC_AclSetConversationID()`.

#### 13.14.2.26 `int fipa_expression_Parse (fipa_expression_t ** expression, fipa_message_string_p message)`

Definition at line 713 of file `fipa_acl.c`.

References `fipa_CheckNextToken()`, `fipa_datetime_Parse()`, `FIPA_EXPR_DATETIME`, `FIPA_EXPR_EXPRESSION`, `FIPA_EXPR_STRING`, `FIPA_EXPR_WORD`, `fipa_expression_Parse()`, `fipa_GetAtom()`, `fipa_string_Parse()`, `fipa_word_Parse()`, `MC_ERR_PARSE`, `MC_SUCCESS`, and `fipa_expression_s::type`.

Referenced by `fipa_expression_Parse()`, and `fipa_message_parameter_Parse()`.

#### 13.14.2.27 `int fipa_GetAtom (fipa_message_string_p message, char expected_atom)`

Definition at line 629 of file `fipa_acl.c`.

References `MC_ERR_PARSE`, `MC_SUCCESS`, and `fipa_message_string_s::parse`.

Referenced by `fipa_acl_Parse()`, `fipa_agent_identifier_Parse()`, `fipa_agent_identifier_set_Parse()`, `fipa_expression_Parse()`, `fipa_message_parameter_Parse()`, `fipa_string_Parse()`, and `fipa_url_sequence_Parse()`.

#### 13.14.2.28 `int fipa_GetNextWord (char ** word, const fipa_message_string_p message)`

Definition at line 764 of file `fipa_acl.c`.

References `ERR`, `MC_SUCCESS`, and `fipa_message_string_s::parse`.

**13.14.2.29 int fipa\_GetWholeToken (char \*\* *word*, fipa\_message\_string\_p *message*)**

Definition at line 812 of file fipa\_acl.c.

References MC\_SUCCESS, and fipa\_message\_string\_s::parse.

Referenced by fipa\_datetime\_Parse().

**13.14.2.30 int fipa\_message\_parameter\_Parse (fipa\_acl\_message\_p *acl*, fipa\_message\_string\_p *message*)**

Definition at line 484 of file fipa\_acl.c.

References fipa\_acl\_message\_s::content, fipa\_word\_s::content, fipa\_acl\_message\_s::conversation\_id, fipa\_acl\_message\_s::encoding, fipa\_agent\_identifier\_Parse(), fipa\_agent\_identifier\_set\_Parse(), fipa\_datetime\_Parse(), fipa\_expression\_Parse(), fipa\_GetAtom(), fipa\_protocol\_type\_Parse(), fipa\_string\_Parse(), fipa\_word\_Destroy(), fipa\_word\_Parse(), fipa\_acl\_message\_s::in\_reply\_to, fipa\_acl\_message\_s::language, MC\_ERR\_PARSE, fipa\_acl\_message\_s::ontology, fipa\_acl\_message\_s::protocol, fipa\_acl\_message\_s::receiver, fipa\_acl\_message\_s::reply\_by, fipa\_acl\_message\_s::reply\_to, fipa\_acl\_message\_s::reply\_with, and fipa\_acl\_message\_s::sender.

Referenced by fipa\_acl\_Parse().

**13.14.2.31 fipa\_message\_string\_t\* fipa\_message\_string\_Copy (fipa\_message\_string\_t \* *src*)**

Definition at line 128 of file fipa\_acl.c.

References fipa\_message\_string\_s::message, and fipa\_message\_string\_s::parse.

**13.14.2.32 int fipa\_message\_string\_Destroy (fipa\_message\_string\_t \* *message*)**

Definition at line 118 of file fipa\_acl.c.

References fipa\_message\_string\_s::message.

Referenced by acc\_connection\_Thread().

**13.14.2.33 fipa\_message\_string\_t\* fipa\_message\_string\_New (void)**

Definition at line 110 of file fipa\_acl.c.

Referenced by acc\_connection\_Thread().

**13.14.2.34 int fipa\_message\_type\_Parse (enum fipa\_performative\_e \* *performative*, fipa\_message\_string\_p *message*)**

Definition at line 567 of file fipa\_acl.c.

References fipa\_word\_s::content, FIPA\_ACCEPT\_PROPOSAL, FIPA\_AGREE, FIPA\_CALL\_FOR\_PROPOSAL, FIPA\_CANCEL, FIPA\_CONFIRM, FIPA\_DISCONFIRM, FIPA\_FAILURE, FIPA\_INFORM, FIPA\_INFORM\_IF, FIPA\_INFORM\_REF, FIPA\_NOT\_UNDERSTOOD, FIPA\_PROPOGATE, FIPA\_PROPOSE, FIPA\_PROXY, FIPA\_QUERY\_IF, FIPA\_QUERY\_REF, FIPA\_REFUSE, FIPA\_REJECT\_PROPOSAL, FIPA\_REQUEST, FIPA\_REQUEST\_WHEN, FIPA\_REQUEST\_WHENEVER, FIPA\_SUBSCRIBE, fipa\_word\_Destroy(), fipa\_word\_Parse(), and MC\_ERR\_PARSE.

Referenced by `fipa_acl_Parse()`.

#### 13.14.2.35 `int fipa_number_Compose (dynstring_t * msg, fipa_number_t * number)`

Definition at line 1547 of file `fipa_acl.c`.

References `dynstring_Append()`, and `fipa_number_s::str`.

Referenced by `fipa_expression_Compose()`.

#### 13.14.2.36 `fipa_number_t* fipa_number_Copy (fipa_number_t * src)`

Definition at line 454 of file `fipa_acl.c`.

References `fipa_number_New()`, and `fipa_number_s::str`.

Referenced by `fipa_expression_Copy()`.

#### 13.14.2.37 `int fipa_number_Destroy (fipa_number_t * number)`

Definition at line 444 of file `fipa_acl.c`.

References `fipa_number_s::str`.

Referenced by `fipa_expression_Destroy()`.

#### 13.14.2.38 `fipa_number_t* fipa_number_New (void)`

Definition at line 436 of file `fipa_acl.c`.

Referenced by `fipa_number_Copy()`.

#### 13.14.2.39 `int fipa_performative_Compose (dynstring_t * msg, enum fipa_performative_e performative)`

Definition at line 1345 of file `fipa_acl.c`.

References `dynstring_Append()`, `FIPA_ACCEPT_PROPOSAL`, `FIPA_AGREE`, `FIPA_CALL_FOR_PROPOSAL`, `FIPA_CANCEL`, `FIPA_CONFIRM`, `FIPA_DISCONFIRM`, `FIPA_FAILURE`, `FIPA_INFORM`, `FIPA_INFORM_IF`, `FIPA_INFORM_REF`, `FIPA_NOT_UNDERSTOOD`, `FIPA_PROPOGATE`, `FIPA_PROPOSE`, `FIPA_PROXY`, `FIPA_QUERY_IF`, `FIPA_QUERY_REF`, `FIPA_REFUSE`, `FIPA_REJECT_PROPOSAL`, `FIPA_REQUEST`, `FIPA_REQUEST_WHEN`, `FIPA_REQUEST_WHENEVER`, `FIPA_SUBSCRIBE`, and `MC_ERR_PARSE`.

Referenced by `fipa_acl_Compose()`.

#### 13.14.2.40 `int fipa_protocol_Compose (dynstring_t * msg, enum fipa_protocol_e protocol)`

Definition at line 1302 of file `fipa_acl.c`.

References `dynstring_Append()`, `FIPA_PROTOCOL_BROKERING`, `FIPA_PROTOCOL_CONTRACT_NET`, `FIPA_PROTOCOL_DUTCH_AUCTION`, `FIPA_PROTOCOL_ENGLISH_AUCTION`, `FIPA_PROTOCOL_ITERATED_CONTRACT_NET`, `FIPA_PROTOCOL_PROPOSE`, `FIPA_PROTOCOL_QUERY`, `FIPA_PROTOCOL_RECRUITING`, `FIPA_PROTOCOL_REQUEST`, `FIPA_PROTOCOL_REQUEST_WHEN`, `FIPA_PROTOCOL_SUBSCRIBE`, and `MC_ERR_PARSE`.

Referenced by `fipa_acl_Compose()`.

#### **13.14.2.41** `int fipa_protocol_type_Parse (enum fipa_protocol_e * protocol, fipa_message_string_p message)`

Definition at line 527 of file `fipa_acl.c`.

References `fipa_word_s::content`, `FIPA_PROTOCOL_BROKERING`, `FIPA_PROTOCOL_CONTRACT_NET`, `FIPA_PROTOCOL_DUTCH_AUCTION`, `FIPA_PROTOCOL_ENGLISH_AUCTION`, `FIPA_PROTOCOL_ITERATED_CONTRACT_NET`, `FIPA_PROTOCOL_PROPOSE`, `FIPA_PROTOCOL_QUERY`, `FIPA_PROTOCOL_RECRUITING`, `FIPA_PROTOCOL_REQUEST`, `FIPA_PROTOCOL_REQUEST_WHEN`, `FIPA_PROTOCOL_SUBSCRIBE`, `fipa_word_Destroy()`, `fipa_word_Parse()`, and `MC_ERR_PARSE`.

Referenced by `fipa_message_parameter_Parse()`.

#### **13.14.2.42** `struct fipa_acl_message_s* fipa_Reply (struct fipa_acl_message_s * acl) [read]`

Definition at line 1555 of file `fipa_acl.c`.

References `fipa_acl_message_s::conversation_id`, `fipa_acl_message_New()`, `fipa_agent_identifier_Copy()`, `fipa_agent_identifier_set_Copy()`, `fipa_agent_identifier_set_New()`, `fipa_agent_identifier_set_s::fipa_agent_identifiers`, `fipa_expression_Copy()`, `fipa_agent_identifier_set_s::num`, `fipa_acl_message_s::protocol`, `fipa_acl_message_s::receiver`, `fipa_acl_message_s::reply_to`, `fipa_agent_identifier_set_s::retain_order`, and `fipa_acl_message_s::sender`.

Referenced by `MC_AclReply()`.

#### **13.14.2.43** `int fipa_string_Compose (dynstring_t * msg, fipa_string_t * string)`

Definition at line 1510 of file `fipa_acl.c`.

References `fipa_string_s::content`, and `dynstring_Append()`.

Referenced by `fipa_acl_Compose()`, and `fipa_expression_Compose()`.

#### **13.14.2.44** `fipa_string_t* fipa_string_Copy (fipa_string_t * src)`

Definition at line 373 of file `fipa_acl.c`.

References `fipa_string_s::content`, and `fipa_string_New()`.

Referenced by `fipa_acl_message_Copy()`, and `fipa_expression_Copy()`.

#### **13.14.2.45** `int fipa_string_Destroy (fipa_string_t * str)`

Definition at line 363 of file `fipa_acl.c`.

References `fipa_string_s::content`.

Referenced by `fipa_acl_message_Destroy()`, `fipa_expression_Destroy()`, and `MC_AclSetContent()`.

#### **13.14.2.46** `fipa_string_t* fipa_string_New (void)`

Definition at line 355 of file `fipa_acl.c`.



Referenced by fipa\_string\_Copy(), MC\_AclSetContent(), and MC\_AclSetConversationID().

#### 13.14.2.47 int fipa\_string\_Parse (fipa\_string\_p \* *fipa\_string*, fipa\_message\_string\_p *message*)

Definition at line 987 of file fipa\_acl.c.

References fipa\_string\_s::content, fipa\_GetAtom(), MC\_ERR\_PARSE, MC\_SUCCESS, and fipa\_message\_string\_s::parse.

Referenced by fipa\_expression\_Parse(), and fipa\_message\_parameter\_Parse().

#### 13.14.2.48 int fipa\_url\_Compose (dynstring\_t \* *msg*, fipa\_url\_t \* *url*)

Definition at line 1539 of file fipa\_acl.c.

References dynstring\_Append(), and fipa\_url\_s::str.

Referenced by fipa\_url\_sequence\_Compose().

#### 13.14.2.49 fipa\_url\_t\* fipa\_url\_Copy (fipa\_url\_t \* *src*)

Definition at line 426 of file fipa\_acl.c.

References fipa\_url\_New(), and fipa\_url\_s::str.

Referenced by fipa\_acl\_envelope\_Received\_Copy(), and fipa\_url\_sequence\_Copy().

#### 13.14.2.50 int fipa\_url\_Destroy (fipa\_url\_t \* *url*)

Definition at line 416 of file fipa\_acl.c.

References fipa\_url\_s::str.

Referenced by fipa\_acl\_envelope\_Received\_Destroy(), and fipa\_url\_sequence\_Destroy().

#### 13.14.2.51 fipa\_url\_t\* fipa\_url\_New (void)

Definition at line 408 of file fipa\_acl.c.

Referenced by fipa\_envelope\_HandleReceived(), fipa\_envelope\_ParseAddresses(), fipa\_url\_Copy(), MC\_AclAddReceiver(), MC\_AclAddReplyTo(), and MC\_AclSetSender().

#### 13.14.2.52 int fipa\_url\_Parse (fipa\_url\_p \* *url*, fipa\_message\_string\_p *message*)

Definition at line 1171 of file fipa\_acl.c.

References fipa\_word\_s::content, fipa\_word\_Destroy(), and fipa\_word\_Parse().

Referenced by fipa\_url\_sequence\_Parse().

#### 13.14.2.53 int fipa\_url\_sequence\_Compose (dynstring\_t \* *msg*, fipa\_url\_sequence\_t \* *urls*)

Definition at line 1420 of file fipa\_acl.c.

References `dynstring_Append()`, `fipa_url_Compose()`, `fipa_url_sequence_s::num`, and `fipa_url_sequence_s::urls`.

Referenced by `fipa_agent_identifier_Compose()`.

#### **13.14.2.54 `fipa_url_sequence_t*` `fipa_url_sequence_Copy` (`fipa_url_sequence_t * src`)**

Definition at line 158 of file `fipa_acl.c`.

References `fipa_url_Copy()`, `fipa_url_sequence_New()`, `fipa_url_sequence_s::num`, and `fipa_url_sequence_s::urls`.

Referenced by `fipa_agent_identifier_Copy()`.

#### **13.14.2.55 `int` `fipa_url_sequence_Destroy` (`fipa_url_sequence_t * sequence`)**

Definition at line 146 of file `fipa_acl.c`.

References `fipa_url_Destroy()`, `fipa_url_sequence_s::num`, and `fipa_url_sequence_s::urls`.

Referenced by `fipa_agent_identifier_Destroy()`.

#### **13.14.2.56 `fipa_url_sequence_t*` `fipa_url_sequence_New` (`void`)**

Definition at line 138 of file `fipa_acl.c`.

Referenced by `fipa_url_sequence_Copy()`, `fipa_url_sequence_Parse()`, `MC_AclAddReceiver()`, `MC_AclAddReplyTo()`, and `MC_AclSetSender()`.

#### **13.14.2.57 `int` `fipa_url_sequence_Parse` (`fipa_url_sequence_p * urls`, `fipa_message_string_p message`)**

Definition at line 1139 of file `fipa_acl.c`.

References `fipa_word_s::content`, `fipa_GetAtom()`, `fipa_url_Parse()`, `fipa_url_sequence_New()`, `fipa_word_Destroy()`, `fipa_word_Parse()`, and `MC_ERR_PARSE`.

Referenced by `fipa_agent_identifier_Parse()`.

#### **13.14.2.58 `int` `fipa_word_Compose` (`dynstring_t * msg`, `fipa_word_t * word`)**

Definition at line 1502 of file `fipa_acl.c`.

References `fipa_word_s::content`, and `dynstring_Append()`.

Referenced by `fipa_expression_Compose()`.

#### **13.14.2.59 `fipa_word_t*` `fipa_word_Copy` (`fipa_word_t * src`)**

Definition at line 345 of file `fipa_acl.c`.

References `fipa_word_s::content`, and `fipa_word_New()`.

Referenced by `fipa_expression_Copy()`.

**13.14.2.60 int fipa\_word\_Destroy (fipa\_word\_t \* *word*)**

Definition at line 335 of file fipa\_acl.c.

References fipa\_word\_s::content.

Referenced by fipa\_agent\_identifier\_Parse(), fipa\_expression\_Destroy(), fipa\_message\_parameter\_Parse(), fipa\_message\_type\_Parse(), fipa\_protocol\_type\_Parse(), fipa\_url\_Parse(), and fipa\_url\_sequence\_Parse().

**13.14.2.61 fipa\_word\_t\* fipa\_word\_New (void)**

Definition at line 327 of file fipa\_acl.c.

Referenced by fipa\_word\_Copy().

**13.14.2.62 int fipa\_word\_Parse (fipa\_word\_t \*\* *word*, fipa\_message\_string\_p *message*)**

Definition at line 652 of file fipa\_acl.c.

References CHECK\_NULL, MC\_ERR\_PARSE, MC\_SUCCESS, and fipa\_message\_string\_s::parse.

Referenced by fipa\_agent\_identifier\_Parse(), fipa\_agent\_identifier\_set\_Parse(), fipa\_expression\_Parse(), fipa\_message\_parameter\_Parse(), fipa\_message\_type\_Parse(), fipa\_protocol\_type\_Parse(), fipa\_url\_Parse(), and fipa\_url\_sequence\_Parse().

## 13.15 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/fipa\_envelope.c File Reference

```
#include "config.h"
#include <stdio.h>
#include <mxml.h>
#include <time.h>
#include "include/fipa_acl_envelope.h"
#include "include/mc_error.h"
#include "include/mc_platform.h"
```

### Functions

- [fipa\\_acl\\_envelope\\_Received\\_t \\* fipa\\_acl\\_envelope\\_Received\\_New](#) (void)
- [int fipa\\_acl\\_envelope\\_Received\\_Destroy](#) (fipa\_acl\_envelope\_Received\_t \*received)
- [fipa\\_acl\\_envelope\\_Received\\_t \\* fipa\\_acl\\_envelope\\_Received\\_Copy](#) (fipa\_acl\_envelope\_Received\_t \*received)
- [fipa\\_acl\\_Param\\_t \\* fipa\\_acl\\_Param\\_New](#) (void)
- [int fipa\\_acl\\_Param\\_Destroy](#) (fipa\_acl\_Param\_t \*param)
- [fipa\\_acl\\_Param\\_t \\* fipa\\_acl\\_Param\\_Copy](#) (fipa\_acl\_Param\_t \*param)
- [fipa\\_acl\\_envelope\\_t \\* fipa\\_acl\\_envelope\\_New](#) (void)
- [int fipa\\_acl\\_envelope\\_Destroy](#) (fipa\_acl\_envelope\_t \*envelope)
- [fipa\\_acl\\_envelope\\_t \\* fipa\\_acl\\_envelope\\_Copy](#) (fipa\_acl\_envelope\_t \*envelope)
- [int fipa\\_envelope\\_Parse](#) (struct fipa\_acl\_envelope\_s \*envelope, const char \*message)
- [int fipa\\_envelope\\_HandleEnvelope](#) (struct fipa\_acl\_envelope\_s \*envelope, mxml\_node\_t \*node)
- [int fipa\\_envelope\\_HandleParams](#) (struct fipa\_acl\_envelope\_s \*envelope, mxml\_node\_t \*node)
- [int fipa\\_envelope\\_HandleTo](#) (struct fipa\_acl\_envelope\_s \*envelope, mxml\_node\_t \*param\_node, int cur\_param)
- [int fipa\\_envelope\\_HandleFrom](#) (struct fipa\_acl\_envelope\_s \*envelope, mxml\_node\_t \*param\_node, int cur\_param)
- [int fipa\\_envelope\\_HandleComments](#) (struct fipa\_acl\_envelope\_s \*envelope, mxml\_node\_t \*param\_node, int cur\_param)
- [int fipa\\_envelope\\_HandleAclRepresentation](#) (struct fipa\_acl\_envelope\_s \*envelope, mxml\_node\_t \*param\_node, int cur\_param)
- [int fipa\\_envelope\\_HandlePayloadLength](#) (struct fipa\_acl\_envelope\_s \*envelope, mxml\_node\_t \*param\_node, int cur\_param)
- [int fipa\\_envelope\\_HandlePayloadEncoding](#) (struct fipa\_acl\_envelope\_s \*envelope, mxml\_node\_t \*param\_node, int cur\_param)
- [int fipa\\_envelope\\_HandleDate](#) (struct fipa\_acl\_envelope\_s \*envelope, mxml\_node\_t \*param\_node, int cur\_param)
- [int fipa\\_envelope\\_HandleIntendedReceiver](#) (struct fipa\_acl\_envelope\_s \*envelope, mxml\_node\_t \*param\_node, int cur\_param)
- [int fipa\\_envelope\\_HandleReceived](#) (struct fipa\_acl\_envelope\_s \*envelope, mxml\_node\_t \*param\_node, int cur\_param)
- [int fipa\\_envelope\\_ParseAgentIdentifier](#) (struct fipa\_agent\_identifier\_s \*\*aid, mxml\_node\_t \*agent\_identifier\_node)
- [int fipa\\_envelope\\_ParseAddresses](#) (struct fipa\_agent\_identifier\_s \*aid, mxml\_node\_t \*addresses\_node)

- `int fipa_envelope_ParseResolvers` (struct `fipa_agent_identifier_s` \*aid, `mxml_node_t` \*resolvers\_node)
- `char * fipa_envelope_Compose` (`fipa_acl_message_t` \*fipa\_acl)
- `mxml_node_t * fipa_envelope_Compose__envelope` (`fipa_acl_message_t` \*fipa\_acl)
- `mxml_node_t * fipa_envelope_Compose__params` (`fipa_acl_message_t` \*fipa\_acl)
- `mxml_node_t * fipa_envelope_Compose__to` (`fipa_acl_message_t` \*fipa\_acl)
- `mxml_node_t * fipa_envelope_Compose__from` (`fipa_acl_message_t` \*fipa\_acl)
- `mxml_node_t * fipa_envelope_Compose__acl_representation` (`fipa_acl_message_t` \*fipa\_acl)
- `mxml_node_t * fipa_envelope_Compose__payload_encoding` (`fipa_acl_message_t` \*fipa\_acl)
- `mxml_node_t * fipa_envelope_Compose__date` (`fipa_acl_message_t` \*fipa\_acl)
- `mxml_node_t * fipa_envelope_Compose__intended_receiver` (`fipa_acl_message_t` \*fipa\_acl)

## 13.15.1 Function Documentation

### 13.15.1.1 `fipa_acl_envelope_t* fipa_acl_envelope_Copy` (`fipa_acl_envelope_t` \* *envelope*)

Definition at line 150 of file `fipa_envelope.c`.

References `fipa_acl_envelope_New()`, `fipa_acl_Param_Copy()`, `fipa_acl_envelope_s::num_params`, and `fipa_acl_envelope_s::params`.

### 13.15.1.2 `int fipa_acl_envelope_Destroy` (`fipa_acl_envelope_t` \* *envelope*)

Definition at line 137 of file `fipa_envelope.c`.

References `fipa_acl_Param_Destroy()`, `fipa_acl_envelope_s::num_params`, and `fipa_acl_envelope_s::params`.

Referenced by `acc_connection_Thread()`.

### 13.15.1.3 `fipa_acl_envelope_t* fipa_acl_envelope_New` (`void`)

Definition at line 129 of file `fipa_envelope.c`.

Referenced by `acc_connection_Thread()`, and `fipa_acl_envelope_Copy()`.

### 13.15.1.4 `fipa_acl_envelope_Received_t* fipa_acl_envelope_Received_Copy` (`fipa_acl_envelope_Received_t` \* *received*)

Definition at line 69 of file `fipa_envelope.c`.

References `fipa_acl_envelope_Received_New()`, `fipa_DateTime_Copy()`, `fipa_url_Copy()`, `fipa_acl_envelope_Received_s::received_by`, `fipa_acl_envelope_Received_s::received_date`, `fipa_acl_envelope_Received_s::received_from`, `fipa_acl_envelope_Received_s::received_id`, and `fipa_acl_envelope_Received_s::received_via`.

Referenced by `fipa_acl_Param_Copy()`.

### 13.15.1.5 `int fipa_acl_envelope_Received_Destroy` (`fipa_acl_envelope_Received_t` \* *received*)

Definition at line 57 of file `fipa_envelope.c`.

References `fipa_DateTime_Destroy()`, `fipa_url_Destroy()`, `fipa_acl_envelope_Received_s::received_by`, `fipa_acl_envelope_Received_s::received_date`, `fipa_acl_envelope_Received_s::received_from`, `fipa_acl_envelope_Received_s::received_id`, and `fipa_acl_envelope_Received_s::received_via`.

Referenced by `fipa_acl_Param_Destroy()`.

#### 13.15.1.6 `fipa_acl_envelope_Received_t* fipa_acl_envelope_Received_New (void)`

Definition at line 48 of file `fipa_envelope.c`.

Referenced by `fipa_acl_envelope_Received_Copy()`, and `fipa_envelope_HandleReceived()`.

#### 13.15.1.7 `fipa_acl_Param_t* fipa_acl_Param_Copy (fipa_acl_Param_t * param)`

Definition at line 110 of file `fipa_envelope.c`.

References `fipa_acl_Param_s::acl_representation`, `fipa_acl_Param_s::comments`, `fipa_acl_Param_s::date`, `fipa_acl_envelope_Received_Copy()`, `fipa_acl_Param_New()`, `fipa_agent_identifier_Copy()`, `fipa_agent_identifier_set_Copy()`, `fipa_DateTime_Copy()`, `fipa_acl_Param_s::from`, `fipa_acl_Param_s::intended_receiver`, `fipa_acl_Param_s::payload_encoding`, `fipa_acl_Param_s::payload_length`, `fipa_acl_Param_s::received`, and `fipa_acl_Param_s::to`.

Referenced by `fipa_acl_envelope_Copy()`.

#### 13.15.1.8 `int fipa_acl_Param_Destroy (fipa_acl_Param_t * param)`

Definition at line 93 of file `fipa_envelope.c`.

References `fipa_acl_Param_s::acl_representation`, `fipa_acl_Param_s::comments`, `fipa_acl_Param_s::date`, `fipa_acl_envelope_Received_Destroy()`, `fipa_agent_identifier_Destroy()`, `fipa_agent_identifier_set_Destroy()`, `fipa_DateTime_Destroy()`, `fipa_acl_Param_s::from`, `fipa_acl_Param_s::intended_receiver`, `fipa_acl_Param_s::payload_encoding`, `fipa_acl_Param_s::payload_length`, `fipa_acl_Param_s::received`, and `fipa_acl_Param_s::to`.

Referenced by `fipa_acl_envelope_Destroy()`.

#### 13.15.1.9 `fipa_acl_Param_t* fipa_acl_Param_New (void)`

Definition at line 85 of file `fipa_envelope.c`.

Referenced by `fipa_acl_Param_Copy()`, and `fipa_envelope_HandleTo()`.

#### 13.15.1.10 `char* fipa_envelope_Compose (fipa_acl_message_t * fipa_acl)`

Definition at line 869 of file `fipa_envelope.c`.

References `fipa_envelope_Compose__envelope()`, `MXML_ADD_AFTER`, `MXML_ADD_TO_PARENT`, `MXML_NO_CALLBACK`, `mxmlAdd()`, `mxmlDelete()`, `mxmlLoadString()`, `mxmlSaveAllocString()`, and `node`.

Referenced by `MC_AclSend()`.

**13.15.1.11 mxml\_node\_t\* fipa\_envelope\_Compose\_\_acl\_representation (fipa\_acl\_message\_t \* *fipa\_acl*)**

Definition at line 1089 of file fipa\_envelope.c.

References mxmlNewElement(), mxmlNewText(), and node.

Referenced by fipa\_envelope\_Compose\_\_params().

**13.15.1.12 mxml\_node\_t\* fipa\_envelope\_Compose\_\_date (fipa\_acl\_message\_t \* *fipa\_acl*)**

Definition at line 1117 of file fipa\_envelope.c.

References buf, mxmlNewElement(), mxmlNewText(), and node.

Referenced by fipa\_envelope\_Compose\_\_params().

**13.15.1.13 mxml\_node\_t\* fipa\_envelope\_Compose\_\_envelope (fipa\_acl\_message\_t \* *fipa\_acl*)**

Definition at line 894 of file fipa\_envelope.c.

References fipa\_envelope\_Compose\_\_params(), MXML\_ADD\_AFTER, MXML\_ADD\_TO\_PARENT, mxmlAdd(), mxmlNewElement(), and node.

Referenced by fipa\_envelope\_Compose().

**13.15.1.14 mxml\_node\_t\* fipa\_envelope\_Compose\_\_from (fipa\_acl\_message\_t \* *fipa\_acl*)**

Definition at line 1021 of file fipa\_envelope.c.

References fipa\_agent\_identifier\_s::addresses, buf, dynstring\_Append(), dynstring\_Destroy(), dynstring\_New(), g\_mc\_platform, mc\_platform\_s::hostname, dynstring\_s::message, mxmlNewElement(), mxmlNewText(), fipa\_agent\_identifier\_s::name, fipa\_url\_sequence\_s::num, mc\_platform\_s::port, fipa\_acl\_message\_s::sender, fipa\_url\_s::str, and fipa\_url\_sequence\_s::urls.

Referenced by fipa\_envelope\_Compose\_\_params().

**13.15.1.15 mxml\_node\_t\* fipa\_envelope\_Compose\_\_intended\_receiver (fipa\_acl\_message\_t \* *fipa\_acl*)**

Definition at line 1146 of file fipa\_envelope.c.

References fipa\_agent\_identifier\_s::addresses, fipa\_agent\_identifier\_set\_s::fipa\_agent\_identifiers, mxmlNewElement(), mxmlNewText(), fipa\_agent\_identifier\_s::name, node, fipa\_url\_sequence\_s::num, fipa\_acl\_message\_s::receiver, fipa\_url\_s::str, and fipa\_url\_sequence\_s::urls.

Referenced by fipa\_envelope\_Compose\_\_params().

**13.15.1.16 mxml\_node\_t\* fipa\_envelope\_Compose\_\_params (fipa\_acl\_message\_t \* *fipa\_acl*)**

Definition at line 914 of file fipa\_envelope.c.

References fipa\_envelope\_Compose\_\_acl\_representation(), fipa\_envelope\_Compose\_\_date(), fipa\_envelope\_Compose\_\_from(), fipa\_envelope\_Compose\_\_intended\_receiver(), fipa\_envelope\_Compose\_\_payload\_encoding(), fipa\_envelope\_Compose\_\_to(), MXML\_ADD\_AFTER, MXML\_ADD\_TO\_PARENT, mxmlAdd(), mxmlElementSetAttr(), mxmlNewElement(), and node.

Referenced by `fipa_envelope_Compose__envelope()`.

#### **13.15.1.17** `mxml_node_t* fipa_envelope_Compose__payload_encoding (fipa_acl_message_t * fipa_acl)`

Definition at line 1103 of file `fipa_envelope.c`.

References `mxmlNewElement()`, `mxmlNewText()`, and `node`.

Referenced by `fipa_envelope_Compose__params()`.

#### **13.15.1.18** `mxml_node_t* fipa_envelope_Compose__to (fipa_acl_message_t * fipa_acl)`

Definition at line 975 of file `fipa_envelope.c`.

References `fipa_agent_identifier_s::addresses`, `fipa_agent_identifier_set_s::fipa_agent_identifiers`, `mxmlNewElement()`, `mxmlNewText()`, `fipa_agent_identifier_s::name`, `node`, `fipa_url_sequence_s::num`, `fipa_agent_identifier_set_s::num`, `fipa_acl_message_s::receiver`, `fipa_url_s::str`, and `fipa_url_sequence_s::urls`.

Referenced by `fipa_envelope_Compose__params()`.

#### **13.15.1.19** `int fipa_envelope_HandleAclRepresentation (struct fipa_acl_envelope_s * envelope, mxml_node_t * param_node, int cur_param)`

Definition at line 381 of file `fipa_envelope.c`.

References `fipa_acl_Param_s::acl_representation`, `mxml_node_s::child`, `MC_ERR_PARSE`, `MXML_DESCEND_FIRST`, `MXML_TEXT`, `mxmlFindElement()`, `fipa_acl_envelope_s::params`, `mxml_text_s::string`, `mxml_value_u::text`, `mxml_node_s::type`, and `mxml_node_s::value`.

Referenced by `fipa_envelope_HandleParams()`.

#### **13.15.1.20** `int fipa_envelope_HandleComments (struct fipa_acl_envelope_s * envelope, mxml_node_t * param_node, int cur_param)`

Definition at line 357 of file `fipa_envelope.c`.

References `mxml_node_s::child`, `fipa_acl_Param_s::comments`, `MC_ERR_PARSE`, `MXML_DESCEND_FIRST`, `MXML_TEXT`, `mxmlFindElement()`, `fipa_acl_envelope_s::params`, `mxml_text_s::string`, `mxml_value_u::text`, `mxml_node_s::type`, and `mxml_node_s::value`.

Referenced by `fipa_envelope_HandleParams()`.

#### **13.15.1.21** `int fipa_envelope_HandleDate (struct fipa_acl_envelope_s * envelope, mxml_node_t * param_node, int cur_param)`

Definition at line 452 of file `fipa_envelope.c`.

References `mxml_node_s::child`, `fipa_acl_Param_s::date`, `fipa_datetime_Parse()`, `MC_ERR_PARSE`, `fipa_message_string_s::message`, `MXML_DESCEND_FIRST`, `MXML_TEXT`, `mxmlFindElement()`, `fipa_acl_envelope_s::params`, `fipa_message_string_s::parse`, `mxml_text_s::string`, `mxml_value_u::text`, `mxml_node_s::type`, and `mxml_node_s::value`.

Referenced by `fipa_envelope_HandleParams()`.



**13.15.1.22 int fipa\_envelope\_HandleEnvelope (struct fipa\_acl\_envelope\_s \* *envelope*, mxml\_node\_t \* *node*)**

Definition at line 178 of file fipa\_envelope.c.

References fipa\_envelope\_HandleParams(), MC\_ERR\_PARSE, MXML\_DESCEND\_FIRST, and mxmlFindElement().

Referenced by fipa\_envelope\_Parse().

**13.15.1.23 int fipa\_envelope\_HandleFrom (struct fipa\_acl\_envelope\_s \* *envelope*, mxml\_node\_t \* *param\_node*, int *cur\_param*)**

Definition at line 328 of file fipa\_envelope.c.

References fipa\_envelope\_ParseAgentIdentifier(), fipa\_acl\_Param\_s::from, MC\_ERR\_PARSE, MXML\_DESCEND\_FIRST, mxmlFindElement(), and fipa\_acl\_envelope\_s::params.

Referenced by fipa\_envelope\_HandleParams().

**13.15.1.24 int fipa\_envelope\_HandleIntendedReceiver (struct fipa\_acl\_envelope\_s \* *envelope*, mxml\_node\_t \* *param\_node*, int *cur\_param*)**

Definition at line 482 of file fipa\_envelope.c.

References fipa\_agent\_identifier\_set\_New(), fipa\_agent\_identifier\_set\_s::fipa\_agent\_identifiers, fipa\_envelope\_ParseAgentIdentifier(), fipa\_acl\_Param\_s::intended\_receiver, MC\_ERR\_PARSE, MXML\_DESCEND\_FIRST, MXML\_NO\_DESCEND, mxmlFindElement(), fipa\_agent\_identifier\_set\_s::num, fipa\_acl\_envelope\_s::params, and fipa\_agent\_identifier\_set\_s::retain\_order.

Referenced by fipa\_envelope\_HandleParams().

**13.15.1.25 int fipa\_envelope\_HandleParams (struct fipa\_acl\_envelope\_s \* *envelope*, mxml\_node\_t \* *node*)**

Definition at line 198 of file fipa\_envelope.c.

References buf, fipa\_envelope\_HandleAclRepresentation(), fipa\_envelope\_HandleComments(), fipa\_envelope\_HandleDate(), fipa\_envelope\_HandleFrom(), fipa\_envelope\_HandleIntendedReceiver(), fipa\_envelope\_HandlePayloadEncoding(), fipa\_envelope\_HandlePayloadLength(), fipa\_envelope\_HandleReceived(), fipa\_envelope\_HandleTo(), MC\_ERR\_PARSE, MXML\_DESCEND\_FIRST, mxmlFindElement(), fipa\_acl\_envelope\_s::num\_params, and fipa\_acl\_envelope\_s::params.

Referenced by fipa\_envelope\_HandleEnvelope().

**13.15.1.26 int fipa\_envelope\_HandlePayloadEncoding (struct fipa\_acl\_envelope\_s \* *envelope*, mxml\_node\_t \* *param\_node*, int *cur\_param*)**

Definition at line 428 of file fipa\_envelope.c.

References mxml\_node\_s::child, MC\_ERR\_PARSE, MXML\_DESCEND\_FIRST, MXML\_TEXT, mxmlFindElement(), fipa\_acl\_envelope\_s::params, fipa\_acl\_Param\_s::payload\_encoding, mxml\_text\_s::string, mxml\_value\_u::text, mxml\_node\_s::type, and mxml\_node\_s::value.

Referenced by fipa\_envelope\_HandleParams().

### 13.15.1.27 **int fipa\_envelope\_HandlePayloadLength** (struct fipa\_acl\_envelope\_s \* *envelope*, mxml\_node\_t \* *param\_node*, int *cur\_param*)

Definition at line 405 of file fipa\_envelope.c.

References mxml\_node\_s::child, MC\_ERR\_PARSE, MXML\_DESCEND\_FIRST, MXML\_TEXT, mxmlFindElement(), fipa\_acl\_envelope\_s::params, fipa\_acl\_Param\_s::payload\_length, mxml\_text\_s::string, mxml\_value\_u::text, mxml\_node\_s::type, and mxml\_node\_s::value.

Referenced by fipa\_envelope\_HandleParams().

### 13.15.1.28 **int fipa\_envelope\_HandleReceived** (struct fipa\_acl\_envelope\_s \* *envelope*, mxml\_node\_t \* *param\_node*, int *cur\_param*)

Definition at line 560 of file fipa\_envelope.c.

References fipa\_acl\_envelope\_Received\_New(), fipa\_datetime\_Parse(), fipa\_url\_New(), MC\_ERR\_PARSE, fipa\_message\_string\_s::message, MXML\_DESCEND\_FIRST, mxmlElementGetAttr(), mxmlFindElement(), node, fipa\_acl\_envelope\_s::params, fipa\_message\_string\_s::parse, fipa\_acl\_Param\_s::received, fipa\_acl\_envelope\_Received\_s::received\_by, fipa\_acl\_envelope\_Received\_s::received\_date, fipa\_acl\_envelope\_Received\_s::received\_from, fipa\_acl\_envelope\_Received\_s::received\_id, fipa\_acl\_envelope\_Received\_s::received\_via, and fipa\_url\_s::str.

Referenced by fipa\_envelope\_HandleParams().

### 13.15.1.29 **int fipa\_envelope\_HandleTo** (struct fipa\_acl\_envelope\_s \* *envelope*, mxml\_node\_t \* *param\_node*, int *cur\_param*)

Definition at line 250 of file fipa\_envelope.c.

References fipa\_acl\_Param\_New(), fipa\_agent\_identifier\_set\_New(), fipa\_agent\_identifier\_set\_s::fipa\_agent\_identifiers, fipa\_envelope\_ParseAgentIdentifier(), MC\_ERR\_PARSE, MXML\_DESCEND\_FIRST, MXML\_NO\_DESCEND, mxmlFindElement(), fipa\_agent\_identifier\_set\_s::num, fipa\_acl\_envelope\_s::params, fipa\_agent\_identifier\_set\_s::retain\_order, and fipa\_acl\_Param\_s::to.

Referenced by fipa\_envelope\_HandleParams().

### 13.15.1.30 **int fipa\_envelope\_Parse** (struct fipa\_acl\_envelope\_s \* *envelope*, const char \* *message*)

Definition at line 165 of file fipa\_envelope.c.

References fipa\_envelope\_HandleEnvelope(), MXML\_NO\_CALLBACK, mxmlDelete(), and mxmlLoadString().

Referenced by acc\_connection\_Thread().

### 13.15.1.31 **int fipa\_envelope\_ParseAddresses** (struct fipa\_agent\_identifier\_s \* *aid*, mxml\_node\_t \* *addresses\_node*)

Definition at line 737 of file fipa\_envelope.c.

References fipa\_agent\_identifier\_s::addresses, mxml\_node\_s::child, fipa\_url\_New(), MC\_ERR\_PARSE, MXML\_DESCEND\_FIRST, MXML\_NO\_DESCEND, MXML\_TEXT, mxmlFindElement(), fipa\_url\_sequence\_s::num, fipa\_url\_s::str, mxml\_text\_s::string, mxml\_value\_u::text, mxml\_node\_s::type, fipa\_url\_sequence\_s::urls, and mxml\_node\_s::value.

Referenced by `fipa_envelope_ParseAgentIdentifier()`.

**13.15.1.32** `int fipa_envelope_ParseAgentIdentifier (struct fipa_agent_identifier_s ** aid, mxml_node_t * agent_identifier_node)`

Definition at line 687 of file `fipa_envelope.c`.

References `mxml_node_s::child`, `fipa_agent_identifier_New()`, `fipa_envelope_ParseAddresses()`, `fipa_envelope_ParseResolvers()`, `MC_ERR_PARSE`, `MXML_DESCEND_FIRST`, `MXML_TEXT`, `mxmlFindElement()`, `mxml_text_s::string`, `mxml_value_u::text`, `mxml_node_s::type`, and `mxml_node_s::value`.

Referenced by `fipa_envelope_HandleFrom()`, `fipa_envelope_HandleIntendedReceiver()`, `fipa_envelope_HandleTo()`, and `fipa_envelope_ParseResolvers()`.

**13.15.1.33** `int fipa_envelope_ParseResolvers (struct fipa_agent_identifier_s * aid, mxml_node_t * resolvers_node)`

Definition at line 804 of file `fipa_envelope.c`.

References `fipa_agent_identifier_set_s::fipa_agent_identifiers`, `fipa_envelope_ParseAgentIdentifier()`, `MC_ERR_PARSE`, `MXML_DESCEND_FIRST`, `MXML_NO_DESCEND`, `mxmlFindElement()`, `fipa_agent_identifier_set_s::num`, `fipa_agent_identifier_s::resolvers`, and `fipa_agent_identifier_set_s::retain_order`.

Referenced by `fipa_envelope_ParseAgentIdentifier()`.

## 13.16 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/acc.h File Reference

```
#include <sys/socket.h>
#include "macros.h"
```

### Functions

- **STRUCT** (acc, struct [mc\\_platform\\_s](#) \*[mc\\_platform](#); [THREAD\\_T](#) [thread](#); [THREAD\\_T](#) [message\\_handler\\_thread](#); [THREAD\\_T](#) [listen\\_thread](#); [THREAD\\_T](#) [udplisten\\_thread](#); [int](#) [waiting](#); [MUTEX\\_T](#) \*[waiting\\_lock](#); [COND\\_T](#) \*[waiting\\_cond](#); [int](#) [num\\_conn\\_threads](#); [THREAD\\_T](#) [conn\\_thread](#); [MUTEX\\_T](#) [conn\\_thread\\_lock](#); [COND\\_T](#) [conn\\_thread\\_cond](#); struct [connection\\_thread\\_arg\\_s](#) \*[connection\\_thread\\_arg](#); [int](#) [num\\_msg\\_threads](#); [THREAD\\_T](#) [msg\\_thread](#); [MUTEX\\_T](#) [msg\\_thread\\_lock](#); [COND\\_T](#) [msg\\_thread\\_cond](#);) **STRUCT**([listen\\_thread\\_arg](#)
- **STRUCT** ([connection\\_thread\\_arg](#), struct [mc\\_platform\\_s](#) \*[mc\\_platform](#); struct [connection\\_s](#) \*[connection](#);) [acc\\_p](#) [acc\\_Initialize](#)(struct [mc\\_platform\\_s](#) \*[mc\\_platform](#))
- [int](#) [acc\\_Destroy](#) ([acc\\_p](#) [acc](#))
- void [acc\\_Start](#) (struct [mc\\_platform\\_s](#) \*[mc\\_platform](#))
- void \* [acc\\_MessageHandlerThread](#) (void \*[arg](#))
- void \* [acc\\_Thread](#) (void \*[arg](#))
- void \* [acc\\_connection\\_Thread](#) (void \*[arg](#))
- void \* [listen\\_Thread](#) (void \*[arg](#))
- void \* [udplisten\\_Thread](#) (void \*[arg](#))

### Variables

- [u\\_long](#) [port](#)
- struct [sockaddr\\_in](#) \* [addr](#)
- unsigned long [int](#) [client\\_fd](#)
- unsigned long [int](#) [server\\_fd](#)

### 13.16.1 Function Documentation

#### 13.16.1.1 void\* acc\_connection\_Thread (void \* arg)

Definition at line 388 of file [acc.c](#).

References [mc\\_platform\\_s::acc](#), [fipa\\_agent\\_identifier\\_s::addresses](#), [agent\\_mailbox\\_Post\(\)](#), [mc\\_platform\\_s::agent\\_queue](#), [AGENT\\_UPDATE](#), [CANCEL](#), [CONNECT\\_THREAD\\_EXIT](#), [connection\\_Destroy\(\)](#), [mtp\\_http\\_s::content](#), [mtp\\_http\\_content\\_s::data](#), [FIPA\\_ACL](#), [fipa\\_acl\\_envelope\\_Destroy\(\)](#), [fipa\\_acl\\_envelope\\_New\(\)](#), [fipa\\_acl\\_message\\_Destroy\(\)](#), [fipa\\_acl\\_message\\_New\(\)](#), [fipa\\_acl\\_Parse\(\)](#), [fipa\\_agent\\_identifier\\_set\\_s::fipa\\_agent\\_identifiers](#), [fipa\\_envelope\\_Parse\(\)](#), [fipa\\_message\\_string\\_Destroy\(\)](#), [fipa\\_message\\_string\\_New\(\)](#), [mtp\\_http\\_s::http\\_performative](#), [HTTP\\_POST](#), [HTTP\\_PUT](#), [agent\\_s::mailbox](#), [agent\\_s::mc\\_platform](#), [mc\\_platform](#), [fipa\\_message\\_string\\_s::message](#), [message\\_s::message\\_body](#), [message\\_Destroy\(\)](#), [message\\_New\(\)](#), [mtp\\_http\\_s::message\\_parts](#), [mc\\_platform\\_s::message\\_queue](#), [message\\_s::message\\_type](#), [message\\_xml\\_parse\(\)](#), [MOBILE\\_AGENT](#), [mtp\\_http\\_Destroy\(\)](#), [mtp\\_http\\_InitializeFromConnection\(\)](#), [mtp\\_http\\_New\(\)](#), [MXML\\_NO\\_CALLBACK](#), [mxmlloadString\(\)](#), [N\\_UNDRSTD](#), [fipa\\_agent\\_identifier\\_s::name](#), [fipa\\_agent\\_identifier\\_set\\_s::num](#), [fipa\\_acl\\_envelope\\_s::num\\_params](#), [fipa\\_acl\\_envelope\\_s::params](#), [fipa\\_message\\_string\\_s::parse](#), [mc\\_platform\\_s::port](#),

mc\_platform\_s::private\_key, QUER\_IF, QUER\_REF, RELAY, REQUEST, RETURN\_MSG, fipa\_url\_s::str, SUBSCRIBE, mtp\_http\_s::target, fipa\_acl\_Param\_s::to, fipa\_url\_sequence\_s::urls, and message\_s::xml\_root.

Referenced by acc\_Thread().

### 13.16.1.2 int acc\_Destroy (acc\_p acc)

Definition at line 93 of file acc.c.

References MC\_SUCCESS.

Referenced by mc\_platform\_Destroy().

### 13.16.1.3 void\* acc\_MessageHandlerThread (void \* arg)

Definition at line 105 of file acc.c.

References mc\_platform\_s::acc, agent\_Initialize(), mc\_platform\_s::agent\_queue, agent\_s::agent\_status, AGENT\_UPDATE, mc\_platform\_s::ams, CANCEL, COND\_BROADCAST, COND\_WAIT, agent\_s::datastate, FIPA\_ACL, mc\_platform\_s::giant, mc\_platform\_s::giant\_cond, mc\_platform\_s::giant\_lock, agent\_s::lock, MC\_AGENT\_NEUTRAL, mc\_platform, MC\_RECV\_AGENT, MC\_RECV\_MESSAGE, MC\_RECV\_RETURN, mc\_platform\_s::MC\_signal, mc\_platform\_s::MC\_signal\_cond, mc\_platform\_s::MC\_signal\_lock, message\_Destroy(), mc\_platform\_s::message\_queue, message\_Send(), message\_s::message\_type, MOBILE\_AGENT, MUTEX\_LOCK, MUTEX\_UNLOCK, N\_UNDRSTD, agent\_s::name, agent\_datastate\_s::persistent, QUER\_IF, QUER\_REF, mc\_platform\_s::quit, mc\_platform\_s::quit\_lock, RELAY, REQUEST, RETURN\_MSG, SUBSCRIBE, THREAD\_EXIT, and message\_s::to\_address.

Referenced by acc\_Start().

### 13.16.1.4 void acc\_Start (struct mc\_platform\_s \* mc\_platform)

### 13.16.1.5 void\* acc\_Thread (void \* arg)

Definition at line 287 of file acc.c.

References mc\_platform\_s::acc, acc\_connection\_Thread(), COND\_BROADCAST, COND\_WAIT, CONN\_THREADS, mc\_platform\_s::connection\_queue, mc\_platform\_s::giant, mc\_platform\_s::giant\_cond, mc\_platform\_s::giant\_lock, mc\_platform, MC\_RECV\_CONNECTION, mc\_platform\_s::MC\_signal, mc\_platform\_s::MC\_signal\_cond, mc\_platform\_s::MC\_signal\_lock, MUTEX\_LOCK, MUTEX\_UNLOCK, mc\_platform\_s::quit, mc\_platform\_s::quit\_lock, THREAD\_CREATE, THREAD\_DETACH, THREAD\_EXIT, and THREAD\_T.

Referenced by acc\_Start().

### 13.16.1.6 void\* listen\_Thread (void \* arg)

Definition at line 652 of file acc.c.

References mc\_platform\_s::acc, connection\_s::addr, connection\_s::AES\_key, mc\_platform\_s::agency, auth\_conn\_rece\_key(), BACKLOG, connection\_s::clientfd, COND\_BROADCAST, connection\_s::connect\_id, connection\_New(), mc\_platform\_s::connection\_queue, agency\_s::known\_host\_filename, mc\_platform, MUTEX\_LOCK, MUTEX\_UNLOCK, connection\_s::nonce, mc\_platform\_s::port, mc\_

platform\_s::private\_key, connection\_s::remote\_hostname, connection\_s::serverfd, SOCKET\_ERROR, mc\_platform\_s::sockfd, and THREAD\_EXIT.

Referenced by acc\_Start().

**13.16.1.7** **STRUCT** (connection\_thread\_arg, struct mc\_platform\_s \*mc\_platform; struct connection\_s \*connection;)

**13.16.1.8** **STRUCT** (acc, struct mc\_platform\_s \*mc\_platform; THREAD\_T thread; THREAD\_T message\_handler\_thread; THREAD\_T listen\_thread; THREAD\_T udplisten\_thread; int waiting; MUTEX\_T \*waiting\_lock; COND\_T \*waiting\_cond; int num\_conn\_threads; THREAD\_T conn\_thread; MUTEX\_T conn\_thread\_lock; COND\_T conn\_thread\_cond; struct connection\_thread\_arg\_s \*connection\_thread\_arg; int num\_msg\_threads; THREAD\_T msg\_thread; MUTEX\_T msg\_thread\_lock; COND\_T msg\_thread\_cond;)

**13.16.1.9** **void\*** udplisten\_Thread (void \*arg)

Definition at line 792 of file acc.c.

References buf, BUFLen, mc\_platform\_s::hostname, mc\_platform, PACKAGE\_VERSION, mc\_platform\_s::port, and UDPPORT.

Referenced by acc\_Start().

## 13.16.2 Variable Documentation

**13.16.2.1** **struct** sockaddr\_in\* addr

Definition at line 78 of file acc.h.

**13.16.2.2** **unsigned long int** client\_fd

Definition at line 79 of file acc.h.

Referenced by main(), and ssl\_test().

**13.16.2.3** **u\_long** port

Definition at line 77 of file acc.h.

Referenced by MC\_AclSend(), MC\_MigrateAgent\_chdl(), MC\_SendAgentMigrationMessage\_chdl(), MC\_SendAgentMigrationMessageFile\_chdl(), message\_send\_Thread(), and message\_xml\_parse\_\_message().

**13.16.2.4** **unsigned long int** server\_fd

Definition at line 80 of file acc.h.

Referenced by main(), and ssl\_test().

## 13.17 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/agent.h File Reference

```
#include "agent_datastate.h"
#include "agent_task.h"
#include "libmc.h"
#include "message.h"
#include "macros.h"
#include "data_structures.h"
#include "agent_mailbox.h"
```

### Data Structures

- struct [agent\\_s](#)

### Typedefs

- typedef struct [agent\\_s](#) [agent\\_t](#)
- typedef [agent\\_t](#) \* [MCAgent\\_t](#)
- typedef [agent\\_t](#) \* [agent\\_p](#)

### Functions

- [int agent\\_AddPersistentVariable](#) ([agent\\_p](#) agent, [int](#) task\_num, const char \*var\_name)
- [agent\\_p agent\\_New](#) (void)
- [agent\\_p agent\\_NewBinary](#) (struct [mc\\_platform\\_s](#) \*mc\_platform)
- [agent\\_p agent\\_Copy](#) (const [agent\\_p](#) agent)
- [int agent\\_Destroy](#) ([agent\\_p](#) agent)
- [agent\\_p agent\\_Initialize](#) (struct [mc\\_platform\\_s](#) \*mc\_platform, [message\\_p](#) message, [int](#) id)
- void [agent\\_RunChScript](#) ([MCAgent\\_t](#) agent, struct [mc\\_platform\\_s](#) \*global)
- void \* [agent\\_RunChScriptThread](#) (void \*agent)
- void \* [agent\\_ChScriptInitVar](#) ([ChInterp\\_t](#) \*interp)

#### 13.17.1 Typedef Documentation

##### 13.17.1.1 typedef [agent\\_t](#)\* [agent\\_p](#)

Definition at line 111 of file agent.h.

##### 13.17.1.2 typedef struct [agent\\_s](#) [agent\\_t](#)

Definition at line 109 of file agent.h.

##### 13.17.1.3 typedef [agent\\_t](#)\* [MCAgent\\_t](#)

Definition at line 110 of file agent.h.

## 13.17.2 Function Documentation

### 13.17.2.1 `int agent_AddPersistentVariable (agent_p agent, int task_num, const char * var_name)`

Definition at line 52 of file agent.c.

References `agent_s::agent_interp`, `agent_task_s::agent_variable_list`, `interpreter_variable_data_s::array_dim`, `interpreter_variable_data_s::array_extent`, `CH_DATATYPE_SIZE`, `CHECK_NULL`, `interpreter_variable_data_s::data`, `interpreter_variable_data_s::data_type`, `agent_s::datastate`, `MC_ERR`, `agent_s::name`, `interpreter_variable_data_s::name`, `interpreter_variable_data_s::size`, `size`, `agent_datastate_s::task_progress`, and `agent_datastate_s::tasks`.

### 13.17.2.2 `void* agent_ChScriptInitVar (ChInterp_t * interp)`

Definition at line 492 of file agent.c.

References `MC_AclAddReceiver_chdl()`, `MC_AclAddReplyTo_chdl()`, `MC_AclDestroy_chdl()`, `MC_AclNew_chdl()`, `MC_AclPost_chdl()`, `MC_AclReply_chdl()`, `MC_AclRetrieve_chdl()`, `MC_AclSend_chdl()`, `MC_AclSetContent_chdl()`, `MC_AclSetConversationID_chdl()`, `MC_AclSetPerformative_chdl()`, `MC_AclSetProtocol_chdl()`, `MC_AclSetSender_chdl()`, `MC_AclWaitRetrieve_chdl()`, `MC_AddAgent_chdl()`, `MC_AgentVariableRetrieve_chdl()`, `MC_AgentVariableSave_chdl()`, `MC_Barrier_chdl()`, `MC_BarrierDelete_chdl()`, `MC_BarrierInit_chdl()`, `MC_CallAgentFunc_chdl()`, `MC_ComposeAgent_chdl()`, `MC_ComposeAgentS_chdl()`, `MC_CondBroadcast_chdl()`, `MC_CondReset_chdl()`, `MC_CondSignal_chdl()`, `MC_CondWait_chdl()`, `MC_DeleteAgent_chdl()`, `MC_DeleteAgentWG_chdl()`, `MC_DeregisterService_chdl()`, `MC_DestroyServiceSearchResult_chdl()`, `MC_End_chdl()`, `MC_FindAgentByID_chdl()`, `MC_FindAgentByName_chdl()`, `MC_GetAgentID_chdl()`, `MC_GetAgentName_chdl()`, `MC_GetAgentStatus_chdl()`, `MC_GetAgentXMLString_chdl()`, `MC_GetTimeOfDay_chdl()`, `MC_HaltAgency_chdl()`, `MC_MigrateAgent_chdl()`, `MC_MutexLock_chdl()`, `MC_MutexUnlock_chdl()`, `MC_PrintAgentCode_chdl()`, `MC_RegisterService_chdl()`, `MC_ResumeAgency_chdl()`, `MC_RetrieveAgent_chdl()`, `MC_RetrieveAgentCode_chdl()`, `MC_SaveData_chdl()`, `MC_SearchForService_chdl()`, `MC_SemaphorePost_chdl()`, `MC_SemaphoreWait_chdl()`, `MC_SendAgentMigrationMessage_chdl()`, `MC_SendAgentMigrationMessageFile_chdl()`, `MC_SendSteerCommand_chdl()`, `MC_SetAgentStatus_chdl()`, `MC_SetDefaultAgentStatus_chdl()`, `MC_SyncDelete_chdl()`, `MC_SyncInit_chdl()`, `MC_TerminateAgent_chdl()`, and `MC_TerminateAgentWG_chdl()`.

Referenced by `AP_QUEUE_STD_DEFN_TEMPLATE()`, and `mc_platform_Initialize()`.

### 13.17.2.3 `agent_p agent_Copy (const agent_p agent)`

Definition at line 134 of file agent.c.

References `agent_datastate_Copy()`, `agent_s::agent_interp`, `agent_mailbox_New()`, `agent_s::agent_persistent`, `agent_s::agent_status`, `agent_s::agent_type`, `agent_s::arrival_time`, `agent_s::datastate`, `agent_s::home`, `agent_s::home_port`, `agent_s::id`, `agent_s::lock`, `agent_s::mailbox`, `MUTEX_INIT`, `MUTEX_LOCK`, `MUTEX_T`, `agent_s::name`, `agent_s::orphan`, `agent_s::owner`, `agent_s::return_data`, and `agent_s::run_lock`.

Referenced by `MC_CopyAgent()`.

### 13.17.2.4 `int agent_Destroy (agent_p agent)`

Definition at line 414 of file agent.c.



References agent\_datastate\_Destroy(), agent\_s::agent\_interp, agent\_mailbox\_Destroy(), agent\_s::agent\_status, agent\_s::datastate, agent\_s::home, mc\_platform\_s::interpreter\_queue, agent\_s::lock, agent\_s::mailbox, MC\_AGENT\_NEUTRAL, agent\_s::mc\_platform, MC\_SUCCESS, MUTEX\_DESTROY, MUTEX\_LOCK, agent\_s::name, agent\_s::owner, agent\_s::run\_lock, agent\_s::sender, and agent\_s::wg\_code.

Referenced by agent\_Initialize().

#### 13.17.2.5 agent\_p agent\_Initialize (struct mc\_platform\_s \* mc\_platform, message\_p message, int id)

Definition at line 293 of file agent.c.

References agent\_s::agent\_address, agent\_datastate\_New(), agent\_Destroy(), agent\_mailbox\_New(), agent\_s::agent\_pipe\_active, agent\_s::agent\_pipe\_ready\_to\_read, agent\_s::agent\_ready\_to\_send, agent\_s::agent\_script\_ready, agent\_s::agent\_status, agent\_s::agent\_thread\_id, agent\_s::agent\_type, message\_s::agent\_xml\_flag, agent\_xml\_parse(), agent\_s::arrival\_time, agent\_s::datastate, mc\_platform\_s::default\_agentstatus, mc\_platform\_s::err, agent\_s::home, mc\_platform\_s::hostname, agent\_s::id, agent\_s::lock, agent\_s::mailbox, MC\_ERR\_PARSE, agent\_s::mc\_platform, MC\_REMOTE\_AGENT, MC\_RETURN\_AGENT, MC\_WAIT\_CH, message\_s::message\_type, MOBILE\_AGENT, MUTEX\_DESTROY, MUTEX\_INIT, MUTEX\_T, agent\_s::orphan, mc\_platform\_s::port, RETURN\_MSG, agent\_s::run\_lock, agent\_s::sender, agent\_datastate\_s::xml\_agent\_root, message\_s::xml\_payload, message\_s::xml\_root, and agent\_datastate\_s::xml\_root.

Referenced by acc\_MessageHandlerThread(), and MC\_SendAgentMigrationMessageFile().

#### 13.17.2.6 agent\_p agent\_New (void)

Definition at line 197 of file agent.c.

References agent\_s::lock, MUTEX\_INIT, MUTEX\_NEW, and agent\_s::run\_lock.

Referenced by MC\_ComposeAgentS().

#### 13.17.2.7 agent\_p agent\_NewBinary (struct mc\_platform\_s \* mc\_platform)

Definition at line 220 of file agent.c.

References agent\_s::agent\_address, agent\_mailbox\_New(), agent\_s::agent\_pipe\_active, agent\_s::agent\_pipe\_ready\_to\_read, agent\_s::agent\_ready\_to\_send, agent\_s::agent\_script\_ready, agent\_s::agent\_status, agent\_s::agent\_thread\_id, agent\_s::arrival\_time, agent\_s::binary, agent\_s::home, mc\_platform\_s::hostname, agent\_s::id, agent\_s::lock, agent\_s::mailbox, MC\_AGENT\_ACTIVE, agent\_s::mc\_platform, MUTEX\_INIT, MUTEX\_T, agent\_s::orphan, mc\_platform\_s::port, agent\_s::run\_lock, and agent\_s::sender.

Referenced by MC\_AddStationaryAgent().

#### 13.17.2.8 void agent\_RunChScript (MCagent\_t agent, struct mc\_platform\_s \* global)

#### 13.17.2.9 void\* agent\_RunChScriptThread (void \* agent)

Definition at line 926 of file agent.c.

References agent\_s::agent\_interp, mc\_platform\_s::ams, COND\_SIGNAL, agent\_s::datastate, agent\_s::id, mc\_platform\_s::interp\_options, mc\_platform\_s::interpreter\_queue, interpreter\_queue\_CreateRetrieve(),

interpreter\_variable\_data\_Destroy(), interpreter\_variable\_data\_Initialize(), interpreter\_variable\_data\_InitializeFromAgent(), MC\_AGENT\_NEUTRAL, MC\_EXEC\_AGENT, agent\_s::mc\_platform, mc\_platform, MC\_RETURN\_AGENT, mc\_platform\_s::MC\_signal, mc\_platform\_s::MC\_signal\_cond, mc\_platform\_s::MC\_signal\_lock, MC\_WAIT\_FINISHED, MC\_WAIT\_MESSGSEND, MUTEX\_LOCK, MUTEX\_UNLOCK, agent\_s::name, SIGNAL, and agent\_datastate\_s::task\_progress.

Referenced by agent\_RunChScript().

## 13.18 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/agent\_datastate.h File Reference

```
#include <mxml.h>
#include "agent_task.h"
```

### Data Structures

- struct [agent\\_datastate\\_s](#)

### Typedefs

- typedef struct [agent\\_datastate\\_s](#) [agent\\_datastate\\_t](#)
- typedef [agent\\_datastate\\_t](#) \* [agent\\_datastate\\_p](#)

### Functions

- [agent\\_datastate\\_p](#) [agent\\_datastate\\_Copy](#) (const [agent\\_datastate\\_p](#) *datastate*)
- [agent\\_datastate\\_p](#) [agent\\_datastate\\_New](#) (void)
- int [agent\\_datastate\\_Destroy](#) ([agent\\_datastate\\_p](#) *agent\_datastate*)

#### 13.18.1 Typedef Documentation

##### 13.18.1.1 typedef [agent\\_datastate\\_t](#)\* [agent\\_datastate\\_p](#)

Definition at line 73 of file [agent\\_datastate.h](#).

##### 13.18.1.2 typedef struct [agent\\_datastate\\_s](#) [agent\\_datastate\\_t](#)

#### 13.18.2 Function Documentation

##### 13.18.2.1 [agent\\_datastate\\_p](#) [agent\\_datastate\\_Copy](#) (const [agent\\_datastate\\_p](#) *datastate*)

Definition at line 47 of file [agent\\_datastate.c](#).

References [agent\\_datastate\\_s::agent\\_code](#), [agent\\_datastate\\_s::agent\\_code\\_ids](#), [agent\\_datastate\\_s::agent\\_codes](#), [agent\\_datastate\\_New\(\)](#), [agent\\_task\\_Copy\(\)](#), [agent\\_datastate\\_s::init\\_agent\\_status](#), [agent\\_datastate\\_s::number\\_of\\_tasks](#), [agent\\_datastate\\_s::persistent](#), [agent\\_datastate\\_s::return\\_data](#), [agent\\_datastate\\_s::task\\_progress](#), and [agent\\_datastate\\_s::tasks](#).

Referenced by [agent\\_Copy\(\)](#).

##### 13.18.2.2 int [agent\\_datastate\\_Destroy](#) ([agent\\_datastate\\_p](#) *agent\_datastate*)

Definition at line 136 of file [agent\\_datastate.c](#).

References [agent\\_datastate\\_s::agent\\_code\\_ids](#), [agent\\_datastate\\_s::agent\\_codes](#), [agent\\_task\\_Destroy\(\)](#), [MC\\_SUCCESS](#), [mxmlDelete\(\)](#), [agent\\_datastate\\_s::number\\_of\\_tasks](#), [agent\\_datastate\\_s::tasks](#), and [agent\\_datastate\\_s::xml\\_root](#).

Referenced by agent\_Destroy().

### 13.18.2.3 agent\_datastate\_p agent\_datastate\_New (void)

Definition at line 115 of file agent\_datastate.c.

References agent\_datastate\_s::agent\_code, CHECK\_NULL, agent\_datastate\_s::init\_agent\_status, agent\_datastate\_s::number\_of\_tasks, agent\_datastate\_s::persistent, agent\_datastate\_s::progress\_modifier, agent\_datastate\_s::return\_data, agent\_datastate\_s::task\_progress, agent\_datastate\_s::tasks, agent\_datastate\_s::xml\_agent\_root, and agent\_datastate\_s::xml\_root.

Referenced by agent\_datastate\_Copy(), agent\_Initialize(), and MC\_ComposeAgentS().

## 13.19 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/agent\_lib.h File Reference

### Functions

- EXPORTCH [int MC\\_AclDestroy\\_chdl](#) (void \*varg)
- EXPORTCH void \* [MC\\_AclNew\\_chdl](#) (void \*varg)
- EXPORTCH void \* [MC\\_AclPost\\_chdl](#) (void \*varg)
- EXPORTCH void \* [MC\\_AclReply\\_chdl](#) (void \*varg)
- EXPORTCH void \* [MC\\_AclRetrieve\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_AclSend\\_chdl](#) (void \*varg)
- EXPORTCH void \* [MC\\_AclWaitRetrieve\\_chdl](#) (void \*varg)
- EXPORTCH void \* [MC\\_AclSetProtocol\\_chdl](#) (void \*varg)
- EXPORTCH void \* [MC\\_AclSetConversationID\\_chdl](#) (void \*varg)
- EXPORTCH void \* [MC\\_AclSetPerformative\\_chdl](#) (void \*varg)
- EXPORTCH void \* [MC\\_AclSetSender\\_chdl](#) (void \*varg)
- EXPORTCH void \* [MC\\_AclAddReceiver\\_chdl](#) (void \*varg)
- EXPORTCH void \* [MC\\_AclAddReplyTo\\_chdl](#) (void \*varg)
- EXPORTCH void \* [MC\\_AclSetContent\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_AddAgent\\_chdl](#) (void \*varg)
- EXPORTCH const void \* [MC\\_AgentVariableRetrieve\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_AgentVariableSave\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_Barrier\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_BarrierDelete\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_BarrierInit\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_CallAgentFunc\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_ComposeAgent\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_ComposeAgentS\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_CondBroadcast\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_CondWait\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_CondReset\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_CondSignal\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_DeleteAgent\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_DeleteAgentWG\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_DeregisterService\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_DestroyServiceSearchResult\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_End\\_chdl](#) (void \*varg)
- EXPORTCH [MC\\_Agent\\_t MC\\_FindAgentByID\\_chdl](#) (void \*varg)
- EXPORTCH [MC\\_Agent\\_t MC\\_FindAgentByName\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_GetAgentID\\_chdl](#) (void \*varg)
- EXPORTCH char \* [MC\\_GetAgentName\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_GetAgentNumTasks\\_chdl](#) (void \*varg)
- EXPORTCH char \* [MC\\_GetAgentXMLString\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_GetTimeOfDay\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_HaltAgency\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_PrintAgentCode\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_MigrateAgent\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_MutexLock\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_MutexUnlock\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_RegisterService\\_chdl](#) (void \*varg)

- EXPORTCH [int MC\\_ResumeAgency\\_chdl](#) (void \*varg)
- EXPORTCH [MC\\_Agent\\_t MC\\_RetrieveAgent\\_chdl](#) (void \*varg)
- EXPORTCH [char \\* MC\\_RetrieveAgentCode\\_chdl](#) (void \*varg)
- EXPORTCH [char \\* MC\\_SearchForService\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_SemaphorePost\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_SemaphoreWait\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_SendSteerCommand\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_TerminateAgent\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_TerminateAgentWG\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_GetAgentStatus\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_SaveData\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_SendAgentMigrationMessage\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_SendAgentMigrationMessageFile\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_SetAgentStatus\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_SetDefaultAgentStatus\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_SyncDelete\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_SyncInit\\_chdl](#) (void \*varg)

### 13.19.1 Function Documentation

#### 13.19.1.1 EXPORTCH void\* MC\_AclAddReceiver\_chdl (void \* *varg*)

Definition at line 2269 of file libmc.c.

References [MC\\_AclAddReceiver\(\)](#).

Referenced by [agent\\_ChScriptInitVar\(\)](#).

#### 13.19.1.2 EXPORTCH void\* MC\_AclAddReplyTo\_chdl (void \* *varg*)

Definition at line 2289 of file libmc.c.

References [MC\\_AclAddReplyTo\(\)](#).

Referenced by [agent\\_ChScriptInitVar\(\)](#).

#### 13.19.1.3 EXPORTCH int MC\_AclDestroy\_chdl (void \* *varg*)

Definition at line 2079 of file libmc.c.

References [MC\\_AclDestroy\(\)](#).

Referenced by [agent\\_ChScriptInitVar\(\)](#).

#### 13.19.1.4 EXPORTCH void\* MC\_AclNew\_chdl (void \* *varg*)

Definition at line 2094 of file libmc.c.

References [MC\\_AclNew\(\)](#).

Referenced by [agent\\_ChScriptInitVar\(\)](#).

**13.19.1.5 EXPORTCH void\* MC\_AclPost\_chdl (void \* *varg*)**

Definition at line 2102 of file libmc.c.

References MC\_AclPost().

Referenced by agent\_ChScriptInitVar().

**13.19.1.6 EXPORTCH void\* MC\_AclReply\_chdl (void \* *varg*)**

Definition at line 2120 of file libmc.c.

References MC\_AclReply().

Referenced by agent\_ChScriptInitVar().

**13.19.1.7 EXPORTCH void\* MC\_AclRetrieve\_chdl (void \* *varg*)**

Definition at line 2136 of file libmc.c.

References MC\_AclRetrieve().

Referenced by agent\_ChScriptInitVar().

**13.19.1.8 EXPORTCH int MC\_AclSend\_chdl (void \* *varg*)**

Definition at line 2152 of file libmc.c.

References CHECK\_NULL, MC\_AclSend(), and agency\_s::mc\_platform.

Referenced by agent\_ChScriptInitVar().

**13.19.1.9 EXPORTCH void\* MC\_AclSetContent\_chdl (void \* *varg*)**

Definition at line 2309 of file libmc.c.

References fipa\_acl\_message\_s::content, and MC\_AclSetContent().

Referenced by agent\_ChScriptInitVar().

**13.19.1.10 EXPORTCH void\* MC\_AclSetConversationID\_chdl (void \* *varg*)**

Definition at line 2213 of file libmc.c.

References MC\_AclSetConversationID().

Referenced by agent\_ChScriptInitVar().

**13.19.1.11 EXPORTCH void\* MC\_AclSetPerformative\_chdl (void \* *varg*)**

Definition at line 2231 of file libmc.c.

References MC\_AclSetPerformative(), and fipa\_acl\_message\_s::performative.

Referenced by agent\_ChScriptInitVar().

**13.19.1.12 EXPORTCH void\* MC\_AclSetProtocol\_chdl (void \* *varg*)**

Definition at line 2194 of file libmc.c.

References MC\_AclSetProtocol(), and fipa\_acl\_message\_s::protocol.

Referenced by agent\_ChScriptInitVar().

**13.19.1.13 EXPORTCH void\* MC\_AclSetSender\_chdl (void \* *varg*)**

Definition at line 2249 of file libmc.c.

References MC\_AclSetSender().

Referenced by agent\_ChScriptInitVar().

**13.19.1.14 EXPORTCH void\* MC\_AclWaitRetrieve\_chdl (void \* *varg*)**

Definition at line 2176 of file libmc.c.

References MC\_AclWaitRetrieve().

Referenced by agent\_ChScriptInitVar().

**13.19.1.15 EXPORTCH int MC\_AddAgent\_chdl (void \* *varg*)**

Definition at line 2329 of file libmc.c.

References CHECK\_NULL, MC\_AddAgent(), and agency\_s::mc\_platform.

Referenced by agent\_ChScriptInitVar().

**13.19.1.16 EXPORTCH const void\* MC\_AgentVariableRetrieve\_chdl (void \* *varg*)**

Definition at line 2353 of file libmc.c.

References MC\_AgentVariableRetrieve().

Referenced by agent\_ChScriptInitVar().

**13.19.1.17 EXPORTCH int MC\_AgentVariableSave\_chdl (void \* *varg*)**

Definition at line 2377 of file libmc.c.

References MC\_AgentVariableSave().

Referenced by agent\_ChScriptInitVar().

**13.19.1.18 EXPORTCH int MC\_Barrier\_chdl (void \* *varg*)**

Definition at line 2429 of file libmc.c.

References CHECK\_NULL, MC\_Barrier(), and agency\_s::mc\_platform.

Referenced by agent\_ChScriptInitVar().



**13.19.1.19 EXPORTCH int MC\_BarrierDelete\_chdl (void \* *varg*)**

Definition at line 2452 of file libmc.c.

References CHECK\_NULL, MC\_BarrierDelete(), and agency\_s::mc\_platform.

Referenced by agent\_ChScriptInitVar().

**13.19.1.20 EXPORTCH int MC\_BarrierInit\_chdl (void \* *varg*)**

Definition at line 2475 of file libmc.c.

References CHECK\_NULL, MC\_BarrierInit(), and agency\_s::mc\_platform.

Referenced by agent\_ChScriptInitVar().

**13.19.1.21 EXPORTCH int MC\_CallAgentFunc\_chdl (void \* *varg*)**

Definition at line 2399 of file libmc.c.

References MC\_CallAgentFuncVar().

Referenced by agent\_ChScriptInitVar().

**13.19.1.22 EXPORTCH int MC\_ComposeAgent\_chdl (void \* *varg*)**

Definition at line 2523 of file libmc.c.

References MC\_ComposeAgentS().

Referenced by agent\_ChScriptInitVar().

**13.19.1.23 EXPORTCH int MC\_ComposeAgentS\_chdl (void \* *varg*)**

Definition at line 2564 of file libmc.c.

References MC\_ComposeAgentS().

Referenced by agent\_ChScriptInitVar().

**13.19.1.24 EXPORTCH int MC\_CondBroadcast\_chdl (void \* *varg*)**

Definition at line 2500 of file libmc.c.

References CHECK\_NULL, MC\_CondBroadcast(), and agency\_s::mc\_platform.

Referenced by agent\_ChScriptInitVar().

**13.19.1.25 EXPORTCH int MC\_CondReset\_chdl (void \* *varg*)**

Definition at line 2630 of file libmc.c.

References CHECK\_NULL, MC\_CondReset(), and agency\_s::mc\_platform.

Referenced by agent\_ChScriptInitVar().

**13.19.1.26 EXPORTCH int MC\_CondSignal\_chdl (void \* *varg*)**

Definition at line 2607 of file libmc.c.

References CHECK\_NULL, MC\_CondSignal(), and agency\_s::mc\_platform.

Referenced by agent\_ChScriptInitVar().

**13.19.1.27 EXPORTCH int MC\_CondWait\_chdl (void \* *varg*)**

Definition at line 2653 of file libmc.c.

References CHECK\_NULL, MC\_CondWait(), and agency\_s::mc\_platform.

Referenced by agent\_ChScriptInitVar().

**13.19.1.28 EXPORTCH int MC\_DeleteAgent\_chdl (void \* *varg*)**

Definition at line 2675 of file libmc.c.

References MC\_DeleteAgent(), MC\_ERR\_NOT\_FOUND, and MC\_FindAgentByName().

Referenced by agent\_ChScriptInitVar().

**13.19.1.29 EXPORTCH int MC\_DeleteAgentWG\_chdl (void \* *varg*)**

Definition at line 2695 of file libmc.c.

References MC\_DeleteAgentWG(), MC\_ERR\_NOT\_FOUND, and MC\_FindAgentByName().

Referenced by agent\_ChScriptInitVar().

**13.19.1.30 EXPORTCH int MC\_DeregisterService\_chdl (void \* *varg*)**

Definition at line 2744 of file libmc.c.

References CHECK\_NULL, MC\_DeregisterService(), and agency\_s::mc\_platform.

Referenced by agent\_ChScriptInitVar().

**13.19.1.31 EXPORTCH int MC\_DestroyServiceSearchResult\_chdl (void \* *varg*)**

Definition at line 2717 of file libmc.c.

References MC\_DestroyServiceSearchResult().

Referenced by agent\_ChScriptInitVar().

**13.19.1.32 EXPORTCH int MC\_End\_chdl (void \* *varg*)**

Definition at line 2771 of file libmc.c.

References CHECK\_NULL, MC\_End(), and agency\_s::mc\_platform.

Referenced by agent\_ChScriptInitVar().

**13.19.1.33 EXPORTCH MCAgent\_t MC\_FindAgentByID\_chdl (void \* *varg*)**

Definition at line 2788 of file libmc.c.

References CHECK\_NULL, MC\_FindAgentByID(), and agency\_s::mc\_platform.

Referenced by agent\_ChScriptInitVar().

**13.19.1.34 EXPORTCH MCAgent\_t MC\_FindAgentByName\_chdl (void \* *varg*)**

Definition at line 2811 of file libmc.c.

References CHECK\_NULL, MC\_FindAgentByName(), and agency\_s::mc\_platform.

Referenced by agent\_ChScriptInitVar().

**13.19.1.35 EXPORTCH int MC\_GetAgentID\_chdl (void \* *varg*)**

Definition at line 2858 of file libmc.c.

References MC\_GetAgentID().

Referenced by agent\_ChScriptInitVar().

**13.19.1.36 EXPORTCH char\* MC\_GetAgentName\_chdl (void \* *varg*)**

Definition at line 2874 of file libmc.c.

References MC\_GetAgentName().

Referenced by agent\_ChScriptInitVar().

**13.19.1.37 EXPORTCH int MC\_GetAgentNumTasks\_chdl (void \* *varg*)**

Definition at line 2890 of file libmc.c.

References MC\_GetAgentNumTasks().

**13.19.1.38 EXPORTCH int MC\_GetAgentStatus\_chdl (void \* *varg*)**

Definition at line 2906 of file libmc.c.

References MC\_GetAgentStatus().

Referenced by agent\_ChScriptInitVar().

**13.19.1.39 EXPORTCH char\* MC\_GetAgentXMLString\_chdl (void \* *varg*)**

Definition at line 2922 of file libmc.c.

References MC\_GetAgentXMLString().

Referenced by agent\_ChScriptInitVar().

**13.19.1.40 EXPORTCH int MC\_GetTimeOfDay\_chdl (void \* *varg*)**

Definition at line 2938 of file libmc.c.

Referenced by agent\_ChScriptInitVar().

**13.19.1.41 EXPORTCH int MC\_HaltAgency\_chdl (void \* *varg*)**

Definition at line 2953 of file libmc.c.

References CHECK\_NULL, MC\_HaltAgency(), and agency\_s::mc\_platform.

Referenced by agent\_ChScriptInitVar().

**13.19.1.42 EXPORTCH int MC\_MigrateAgent\_chdl (void \* *varg*)**

Definition at line 2970 of file libmc.c.

References MC\_MigrateAgent(), and port.

Referenced by agent\_ChScriptInitVar().

**13.19.1.43 EXPORTCH int MC\_MutexLock\_chdl (void \* *varg*)**

Definition at line 2989 of file libmc.c.

References CHECK\_NULL, MC\_MutexLock(), and agency\_s::mc\_platform.

Referenced by agent\_ChScriptInitVar().

**13.19.1.44 EXPORTCH int MC\_MutexUnlock\_chdl (void \* *varg*)**

Definition at line 3013 of file libmc.c.

References CHECK\_NULL, MC\_MutexUnlock(), and agency\_s::mc\_platform.

Referenced by agent\_ChScriptInitVar().

**13.19.1.45 EXPORTCH int MC\_PrintAgentCode\_chdl (void \* *varg*)**

Definition at line 3037 of file libmc.c.

References MC\_PrintAgentCode().

Referenced by agent\_ChScriptInitVar().

**13.19.1.46 EXPORTCH int MC\_RegisterService\_chdl (void \* *varg*)**

Definition at line 3053 of file libmc.c.

References CHECK\_NULL, agency\_s::mc\_platform, and MC\_RegisterService().

Referenced by agent\_ChScriptInitVar().

**13.19.1.47 EXPORTCH int MC\_ResumeAgency\_chdl (void \* *varg*)**

Definition at line 3090 of file libmc.c.

References CHECK\_NULL, agency\_s::mc\_platform, and MC\_ResumeAgency().

Referenced by agent\_ChScriptInitVar().

**13.19.1.48 EXPORTCH MCAgent\_t MC\_RetrieveAgent\_chdl (void \* *varg*)**

Definition at line 3108 of file libmc.c.

References CHECK\_NULL, agency\_s::mc\_platform, and MC\_RetrieveAgent().

Referenced by agent\_ChScriptInitVar().

**13.19.1.49 EXPORTCH char\* MC\_RetrieveAgentCode\_chdl (void \* *varg*)**

Definition at line 3125 of file libmc.c.

References MC\_RetrieveAgentCode().

Referenced by agent\_ChScriptInitVar().

**13.19.1.50 EXPORTCH int MC\_SaveData\_chdl (void \* *varg*)**

Definition at line 3141 of file libmc.c.

References agent\_task\_s::agent\_variable\_list, interpreter\_variable\_data\_s::data, interpreter\_variable\_data\_s::data\_type, agent\_s::datastate, interpreter\_variable\_data\_New(), interpreter\_variable\_data\_s::name, interpreter\_variable\_data\_s::size, size, agent\_datastate\_s::task\_progress, and agent\_datastate\_s::tasks.

Referenced by agent\_ChScriptInitVar().

**13.19.1.51 EXPORTCH char\* MC\_SearchForService\_chdl (void \* *varg*)**

Definition at line 3176 of file libmc.c.

References CHECK\_NULL, agency\_s::mc\_platform, and MC\_SearchForService().

Referenced by agent\_ChScriptInitVar().

**13.19.1.52 EXPORTCH int MC\_SemaphorePost\_chdl (void \* *varg*)**

Definition at line 3217 of file libmc.c.

References CHECK\_NULL, agency\_s::mc\_platform, and MC\_SemaphorePost().

Referenced by agent\_ChScriptInitVar().

**13.19.1.53 EXPORTCH int MC\_SemaphoreWait\_chdl (void \* *varg*)**

Definition at line 3241 of file libmc.c.

References CHECK\_NULL, agency\_s::mc\_platform, and MC\_SemaphoreWait().

Referenced by agent\_ChScriptInitVar().

**13.19.1.54 EXPORTCH int MC\_SendAgentMigrationMessage\_chdl (void \* *varg*)**

Definition at line 3265 of file libmc.c.

References CHECK\_NULL, agency\_s::mc\_platform, MC\_SendAgentMigrationMessage(), and port.

Referenced by agent\_ChScriptInitVar().

**13.19.1.55 EXPORTCH int MC\_SendAgentMigrationMessageFile\_chdl (void \* *varg*)**

Definition at line 3290 of file libmc.c.

References MC\_SendAgentMigrationMessageFile(), and port.

Referenced by agent\_ChScriptInitVar().

**13.19.1.56 EXPORTCH int MC\_SendSteerCommand\_chdl (void \* *varg*)**

Definition at line 3310 of file libmc.c.

References CHECK\_NULL, agency\_s::mc\_platform, and MC\_SendSteerCommand().

Referenced by agent\_ChScriptInitVar().

**13.19.1.57 EXPORTCH int MC\_SetAgentStatus\_chdl (void \* *varg*)**

Definition at line 3333 of file libmc.c.

References MC\_SetAgentStatus().

Referenced by agent\_ChScriptInitVar().

**13.19.1.58 EXPORTCH int MC\_SetDefaultAgentStatus\_chdl (void \* *varg*)**

Definition at line 3351 of file libmc.c.

References CHECK\_NULL, agency\_s::mc\_platform, and MC\_SetDefaultAgentStatus().

Referenced by agent\_ChScriptInitVar().

**13.19.1.59 EXPORTCH int MC\_SyncDelete\_chdl (void \* *varg*)**

Definition at line 3374 of file libmc.c.

References CHECK\_NULL, agency\_s::mc\_platform, and MC\_SyncDelete().

Referenced by agent\_ChScriptInitVar().

**13.19.1.60 EXPORTCH int MC\_SyncInit\_chdl (void \* *varg*)**

Definition at line 3397 of file libmc.c.

References CHECK\_NULL, agency\_s::mc\_platform, and MC\_SyncInit().

Referenced by agent\_ChScriptInitVar().

**13.19.1.61 EXPORTCH int MC\_TerminateAgent\_chdl (void \* *varg*)**

Definition at line 3421 of file libmc.c.

References MC\_ERR\_NOT\_FOUND, MC\_FindAgentByName(), and MC\_TerminateAgent().

Referenced by agent\_ChScriptInitVar().

**13.19.1.62 EXPORTCH int MC\_TerminateAgentWG\_chdl (void \* *varg*)**

Definition at line 3442 of file libmc.c.

References MC\_ERR\_NOT\_FOUND, MC\_FindAgentByName(), and MC\_TerminateAgentWG().

Referenced by agent\_ChScriptInitVar().

## 13.20 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/agent\_mailbox.h File Reference

### Data Structures

- struct [agent\\_mailbox\\_s](#)

### Typedefs

- typedef struct [agent\\_mailbox\\_s](#) [agent\\_mailbox\\_t](#)
- typedef [agent\\_mailbox\\_t](#) \* [agent\\_mailbox\\_p](#)

### Functions

- [agent\\_mailbox\\_p](#) [agent\\_mailbox\\_New](#) (void)
- [agent\\_mailbox\\_p](#) [agent\\_mailbox\\_Copy](#) ([agent\\_mailbox\\_p](#) src)
- int [agent\\_mailbox\\_Destroy](#) ([agent\\_mailbox\\_t](#) \*mailbox)
- int [agent\\_mailbox\\_Post](#) ([agent\\_mailbox\\_p](#) mailbox, [fipa\\_acl\\_message\\_t](#) \*message)
- [fipa\\_acl\\_message\\_t](#) \* [agent\\_mailbox\\_Retrieve](#) ([agent\\_mailbox\\_p](#) mailbox)
- [fipa\\_acl\\_message\\_t](#) \* [agent\\_mailbox\\_WaitRetrieve](#) ([agent\\_mailbox\\_p](#) mailbox)

### 13.20.1 Typedef Documentation

#### 13.20.1.1 typedef [agent\\_mailbox\\_t](#)\* [agent\\_mailbox\\_p](#)

Definition at line 12 of file [agent\\_mailbox.h](#).

#### 13.20.1.2 typedef struct [agent\\_mailbox\\_s](#) [agent\\_mailbox\\_t](#)

### 13.20.2 Function Documentation

#### 13.20.2.1 [agent\\_mailbox\\_p](#) [agent\\_mailbox\\_Copy](#) ([agent\\_mailbox\\_p](#) src)

Definition at line 21 of file [agent\\_mailbox.c](#).

References [agent\\_mailbox\\_New\(\)](#), and [agent\\_mailbox\\_s::mail\\_queue](#).

#### 13.20.2.2 int [agent\\_mailbox\\_Destroy](#) ([agent\\_mailbox\\_t](#) \* mailbox)

Definition at line 29 of file [agent\\_mailbox.c](#).

References [agent\\_mailbox\\_s::mail\\_queue](#).

Referenced by [agent\\_Destroy\(\)](#).

#### 13.20.2.3 [agent\\_mailbox\\_p](#) [agent\\_mailbox\\_New](#) (void)

Definition at line 12 of file [agent\\_mailbox.c](#).

References [agent\\_mailbox\\_s::mail\\_queue](#).



Referenced by agent\_Copy(), agent\_Initialize(), agent\_mailbox\_Copy(), and agent\_NewBinary().

**13.20.2.4 int agent\_mailbox\_Post (agent\_mailbox\_p *mailbox*, fipa\_acl\_message\_t \* *message*)**

Definition at line 38 of file agent\_mailbox.c.

References agent\_mailbox\_s::mail\_queue.

Referenced by acc\_connection\_Thread(), and MC\_AclPost().

**13.20.2.5 fipa\_acl\_message\_t\* agent\_mailbox\_Retrieve (agent\_mailbox\_p *mailbox*)**

Definition at line 44 of file agent\_mailbox.c.

References agent\_mailbox\_s::mail\_queue.

Referenced by agent\_mailbox\_WaitRetrieve(), and MC\_AclRetrieve().

**13.20.2.6 fipa\_acl\_message\_t\* agent\_mailbox\_WaitRetrieve (agent\_mailbox\_p *mailbox*)**

Definition at line 49 of file agent\_mailbox.c.

References agent\_mailbox\_Retrieve(), COND\_WAIT, agent\_mailbox\_s::mail\_queue, MUTEX\_LOCK, and MUTEX\_UNLOCK.

Referenced by MC\_AclWaitRetrieve().

## 13.21 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/agent\_task.h File Reference

```
#include "interpreter_variable_data.h"
#include "data_structures.h"
```

### Data Structures

- struct [agent\\_task\\_s](#)

### Typedefs

- typedef struct [agent\\_task\\_s](#) [agent\\_task\\_t](#)
- typedef [agent\\_task\\_t](#) \* [agent\\_task\\_p](#)

### Functions

- [agent\\_task\\_p](#) [agent\\_task\\_New](#) (void)
- [agent\\_task\\_p](#) [agent\\_task\\_Copy](#) ([agent\\_task\\_p](#) task)
- int [agent\\_task\\_Destroy](#) ([agent\\_task\\_p](#) agent\_task)

#### 13.21.1 Typedef Documentation

##### 13.21.1.1 typedef [agent\\_task\\_t](#)\* [agent\\_task\\_p](#)

Definition at line 59 of file [agent\\_task.h](#).

##### 13.21.1.2 typedef struct [agent\\_task\\_s](#) [agent\\_task\\_t](#)

#### 13.21.2 Function Documentation

##### 13.21.2.1 [agent\\_task\\_p](#) [agent\\_task\\_Copy](#) ([agent\\_task\\_p](#) task)

Definition at line 66 of file [agent\\_task.c](#).

References [agent\\_task\\_s::agent\\_return\\_data](#), [agent\\_task\\_s::agent\\_variable\\_list](#), [agent\\_task\\_s::code\\_id](#), [agent\\_task\\_s::init\\_agent\\_status](#), [interpreter\\_variable\\_data\\_Copy\(\)](#), [ListAdd\(\)](#), [ListSearch\(\)](#), [agent\\_task\\_s::num\\_saved\\_variables](#), [agent\\_task\\_s::number\\_of\\_elements](#), [agent\\_task\\_s::persistent](#), [agent\\_task\\_s::saved\\_variables](#), [agent\\_task\\_s::server\\_name](#), [agent\\_task\\_s::size\\_of\\_element\\_array](#), and [agent\\_task\\_s::var\\_name](#).

Referenced by [agent\\_datastate\\_Copy\(\)](#).

##### 13.21.2.2 int [agent\\_task\\_Destroy](#) ([agent\\_task\\_p](#) agent\_task)

Definition at line 132 of file [agent\\_task.c](#).

### **13.21 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/agent\_task.h File Reference**

References `agent_task_s::agent_return_data`, `agent_task_s::agent_variable_list`, `agent_task_s::code_id`, `interpreter_variable_data_Destroy()`, `MC_SUCCESS`, `agent_task_s::saved_variables`, `agent_task_s::server_name`, and `agent_task_s::var_name`.

Referenced by `agent_datastate_Destroy()`.

#### **13.21.2.3 `agent_task_p agent_task_New (void)`**

Definition at line 48 of file `agent_task.c`.

References `agent_task_s::agent_variable_list`, `agent_task_s::num_saved_variables`, and `agent_task_s::saved_variables`.

Referenced by `agent_xml_parse__tasks()`, and `MC_ComposeAgentS()`.

## 13.22 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/ams.h File Reference

```
#include "macros.h"
```

### Functions

- **STRUCT** (ams, struct [mc\\_platform\\_s](#) \*[mc\\_platform](#);MUTEX\_T \*[runflag\\_lock](#);COND\_T \*[runflag\\_cond](#);int [run](#);int [waiting](#);MUTEX\_T \*[waiting\\_lock](#);COND\_T \*[waiting\\_cond](#);THREAD\_T [thread](#);) [ams\\_p](#) [ams\\_Initialize](#)(struct [mc\\_platform\\_s](#) \*\_[mc\\_platform](#))
- int [ams\\_Destroy](#) ([ams\\_p](#) [ams](#))
- int [ams\\_RemoveFinishedAgents](#) ([ams\\_p](#) [ams](#))
- int [ams\\_ManageAgentList](#) ([ams\\_p](#) [ams](#))
- void [ams\\_Print](#) ([ams\\_p](#) [ams](#))
- void [ams\\_Start](#) (struct [mc\\_platform\\_s](#) \*[mc\\_platform](#))
- void \* [ams\\_Thread](#) (void \*[arg](#))

### 13.22.1 Function Documentation

#### 13.22.1.1 int [ams\\_Destroy](#) ([ams\\_p](#) [ams](#))

Definition at line 46 of file [ams.c](#).

References [COND\\_DESTROY](#), [MC\\_SUCCESS](#), and [MUTEX\\_DESTROY](#).

Referenced by [mc\\_platform\\_Destroy](#)().

#### 13.22.1.2 int [ams\\_ManageAgentList](#) ([ams\\_p](#) [ams](#))

Definition at line 116 of file [ams.c](#).

References [agent\\_RunChScript](#)(), [agent\\_s::agent\\_status](#), [agent\\_s::binary](#), [ListSearch](#)(), [agent\\_s::lock](#), [MC\\_AGENT\\_ACTIVE](#), [MC\\_AGENT\\_NEUTRAL](#), [MC\\_TerminateAgent](#)(), [MC\\_WAIT\\_CH](#), [MC\\_WAIT\\_FINISHED](#), [MC\\_WAIT\\_MESSGSEND](#), [message\\_Destroy](#)(), [message\\_InitializeFromAgent](#)(), [message\\_New](#)(), [MUTEX\\_LOCK](#), [MUTEX\\_UNLOCK](#), [agent\\_s::name](#), [agent\\_s::orphan](#), [mc\\_platform\\_s::quit](#), [mc\\_platform\\_s::quit\\_lock](#), and [agent\\_s::run\\_lock](#).

Referenced by [ams\\_Thread](#)().

#### 13.22.1.3 void [ams\\_Print](#) ([ams\\_p](#) [ams](#))

Definition at line 84 of file [ams.c](#).

References [agent\\_s::agent\\_status](#), [agent\\_s::connect\\_id](#), [agent\\_s::id](#), [ListSearch](#)(), [MUTEX\\_LOCK](#), and [MUTEX\\_UNLOCK](#).

**13.22.1.4** `int ams_RemoveFinishedAgents (ams_p ams)`

**13.22.1.5** `void ams_Start (struct mc_platform_s * mc_platform)`

**13.22.1.6** `void* ams_Thread (void * arg)`

Definition at line 254 of file ams.c.

References `mc_platform_s::ams`, `ams_ManageAgentList()`, `COND_BROADCAST`, `COND_WAIT`, `mc_platform`, `MUTEX_LOCK`, `MUTEX_UNLOCK`, `mc_platform_s::quit`, `mc_platform_s::quit_lock`, and `THREAD_EXIT`.

Referenced by `ams_Start()`.

**13.22.1.7** `STRUCT (ams, struct mc_platform_s *mc_platform;MUTEX_T *runflag_lock;COND_T *runflag_cond;int run;int waiting;MUTEX_T *waiting_lock;COND_T *waiting_cond;THREAD_T thread;)`

## 13.23 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/ap\_queue\_template.h File Reference

```
#include "macros.h"
#include "mc_error.h"
#include <embedch.h>
```

### Data Structures

- struct [AP\\_GENERIC\\_s](#)

### Defines

- #define [AP\\_QUEUE\\_DECL\\_TEMPLATE](#)(name, node\_type)
- #define [AP\\_QUEUE\\_GENERIC\\_DECL\\_TEMPLATE](#)(name, func\_name, return\_type, search\_type) return\_type name##\_##\_func\_name(name##\_p name, const search\_type key);
- #define [AP\\_QUEUE\\_STD\\_DEFN\\_TEMPLATE](#)(name, node\_type)
- #define [AP\\_QUEUE\\_SEARCH\\_TEMPLATE](#)(name, func\_name, node\_type, search\_type, search\_expression)
- #define [AP\\_QUEUE\\_REMOVE\\_TEMPLATE](#)(name, func\_name, node\_type, search\_type, search\_expression)

### Typedefs

- typedef struct [AP\\_GENERIC\\_s](#) [AP\\_GENERIC\\_t](#)
- typedef [AP\\_GENERIC\\_t](#) \* [AP\\_GENERIC\\_p](#)

### Functions

- [AP\\_GENERIC\\_p](#) [interpreter\\_queue\\_CreateRetrieve](#) (struct [interpreter\\_queue\\_s](#) \*queue, [ChOptions\\_t](#) \*interp\_options)

#### 13.23.1 Define Documentation

##### 13.23.1.1 #define AP\_QUEUE\_DECL\_TEMPLATE(name, node\_type)

##### Value:

```
typedef struct name##_s \
{ \
    int size; \
    list_p list; \
    MUTEX_T* lock; \
    COND_T* cond; \
} name##_t; \
\
typedef name##_t* name##_p; \
\
name##_p name##_New( void ); \
int name##_Destroy( name##_p name ); \
```

```

int name##_Add( name##_p name, struct node_type##_s* node ); \
name##_p name##_Copy( name##_p name ); \
struct node_type##_s* name##_Pop( name##_p name ); \
struct node_type##_s* name##_WaitPop( name##_p name ); \
struct node_type##_s* name##_SearchIndex( name##_p name, int index ); \
int name##_RemoveIndex( name##_p name, int index );

```

Definition at line 46 of file ap\_queue\_template.h.

### 13.23.1.2 **#define AP\_QUEUE\_GENERIC\_DECL\_TEMPLATE(name, func\_name, return\_type, search\_type) return\_type name##\_##func\_name(name##\_p name, const search\_type key);**

Definition at line 66 of file ap\_queue\_template.h.

### 13.23.1.3 **#define AP\_QUEUE\_REMOVE\_TEMPLATE(name, func\_name, node\_type, search\_type, search\_expression)**

**Value:**

```

int name##_##func_name( name##_p name, const search_type key ) \
{ \
    int err_code = MC_ERR_NOT_FOUND; \
    struct listNode_s* parsenode; \
    struct node_type##_s* node; \
    node = NULL; \
    \
    MUTEX_LOCK(name->lock); \
    if (name->list->listhead == NULL) { \
        MUTEX_UNLOCK(name->lock); \
        return MC_ERR_NOT_FOUND; \
    } \
    for( \
        parsenode = (listNode_t*)name->list->listhead; \
        parsenode->next != NULL; \
        parsenode = (listNode_t*)parsenode->next \
    ) \
    { \
        node = (node_type##_t*)parsenode->node_data; \
        if (search_expression) { \
            break; \
            err_code = MC_SUCCESS; \
        } \
    } \
    MUTEX_UNLOCK(name->lock); \
    return err_code; \
}

```

Definition at line 224 of file ap\_queue\_template.h.

### 13.23.1.4 **#define AP\_QUEUE\_SEARCH\_TEMPLATE(name, func\_name, node\_type, search\_type, search\_expression)**

**Value:**

```

struct node_type##_s* name##_##func_name( name##_p name, const search_type key ) \
{ \
    listNode_t* parsenode; \

```

```

struct node_type##_s* node; \
struct node_type##_s* ret = NULL; \
node = NULL; \
\
MUTEX_LOCK(name->lock); \
if (name->list->listhead == NULL) { \
    MUTEX_UNLOCK(name->lock); \
    return NULL; \
} \
for( \
    parsenode = (listNode_t*)name->list->listhead; \
    parsenode != NULL; \
    parsenode = (listNode_t*)parsenode->next \
) \
{ \
    node = (node_type##_t*)parsenode->node_data; \
    if (search_expression){ \
        ret = node; \
        break; \
    } \
} \
MUTEX_UNLOCK(name->lock); \
return ret; \
}

```

Definition at line 194 of file ap\_queue\_template.h.

### 13.23.1.5 #define AP\_QUEUE\_STD\_DEFN\_TEMPLATE(name, node\_type)

Definition at line 69 of file ap\_queue\_template.h.

## 13.23.2 Typedef Documentation

### 13.23.2.1 typedef AP\_GENERIC\_t\* AP\_GENERIC\_p

Definition at line 43 of file ap\_queue\_template.h.

### 13.23.2.2 typedef struct AP\_GENERIC\_s AP\_GENERIC\_t

Definition at line 42 of file ap\_queue\_template.h.

## 13.23.3 Function Documentation

### 13.23.3.1 AP\_GENERIC\_p interpreter\_queue\_CreateRetrieve (struct interpreter\_queue\_s \* queue, ChOptions\_t \* interp\_options)

Referenced by agent\_RunChScriptThread().



## 13.24 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/barrier.h File Reference

```
#include "macros.h"
#include "../mc_list/list.h"
#include "mc_rwlock.h"
```

### Data Structures

- struct [barrier\\_node\\_s](#)
- struct [barrier\\_queue\\_s](#)

### Typedefs

- typedef struct [barrier\\_node\\_s](#) [barrier\\_node\\_t](#)
- typedef [barrier\\_node\\_t](#) \* [barrier\\_node\\_p](#)
- typedef struct [barrier\\_queue\\_s](#) [barrier\\_queue\\_t](#)
- typedef [barrier\\_queue\\_t](#) \* [barrier\\_queue\\_p](#)

### Functions

- [barrier\\_node\\_p](#) [barrier\\_node\\_Initialize](#) ([int](#) id, [int](#) num\_registered)
- [int](#) [barrier\\_node\\_Destroy](#) ([barrier\\_node\\_p](#) node)
- [int](#) [barrier\\_queue\\_Add](#) ([barrier\\_queue\\_p](#) list, [barrier\\_node\\_p](#) node)
- [int](#) [barrier\\_queue\\_Delete](#) ([int](#) id, [barrier\\_queue\\_p](#) list)
- [int](#) [barrier\\_queue\\_Destroy](#) ([barrier\\_queue\\_p](#) queue)
- [barrier\\_node\\_p](#) [barrier\\_queue\\_Get](#) ([barrier\\_queue\\_p](#) list, [int](#) id)
- [barrier\\_queue\\_p](#) [barrier\\_queue\\_New](#) ([void](#))
- [barrier\\_node\\_p](#) [barrier\\_queue\\_Pop](#) ([barrier\\_queue\\_p](#) queue)

#### 13.24.1 Typedef Documentation

##### 13.24.1.1 typedef [barrier\\_node\\_t](#)\* [barrier\\_node\\_p](#)

Definition at line 49 of file barrier.h.

##### 13.24.1.2 typedef struct [barrier\\_node\\_s](#) [barrier\\_node\\_t](#)

##### 13.24.1.3 typedef [barrier\\_queue\\_t](#)\* [barrier\\_queue\\_p](#)

Definition at line 57 of file barrier.h.

#### 13.24.1.4 typedef struct barrier\_queue\_s barrier\_queue\_t

### 13.24.2 Function Documentation

#### 13.24.2.1 int barrier\_node\_Destroy (barrier\_node\_p node)

Definition at line 70 of file barrier.c.

References barrier\_node\_s::cond, COND\_DESTROY, barrier\_node\_s::lock, MC\_SUCCESS, and MUTEX\_DESTROY.

Referenced by barrier\_queue\_Delete(), and barrier\_queue\_Destroy().

#### 13.24.2.2 barrier\_node\_p barrier\_node\_Initialize (int id, int num\_registered)

Definition at line 45 of file barrier.c.

References CHECK\_NULL, barrier\_node\_s::cond, COND\_INIT, COND\_T, barrier\_node\_s::id, barrier\_node\_s::lock, MUTEX\_INIT, MUTEX\_T, node, barrier\_node\_s::num\_registered, and barrier\_node\_s::num\_waiting.

Referenced by MC\_BarrierInit().

#### 13.24.2.3 int barrier\_queue\_Add (barrier\_queue\_p list, barrier\_node\_p node)

Definition at line 87 of file barrier.c.

References DATA, barrier\_node\_s::id, barrier\_queue\_s::list, ListAdd(), list\_s::listhead, barrier\_queue\_s::lock, MC\_SUCCESS, MC\_WARN\_DUPLICATE, listNode\_s::next, listNode\_s::node\_data, RWLOCK\_WRLock, RWLOCK\_WRunLock, and barrier\_queue\_s::size.

Referenced by MC\_BarrierInit().

#### 13.24.2.4 int barrier\_queue\_Delete (int id, barrier\_queue\_p list)

Definition at line 111 of file barrier.c.

References barrier\_node\_Destroy(), barrier\_node\_s::id, barrier\_queue\_s::list, ListDelete(), ListSearch(), barrier\_queue\_s::lock, MC\_ERR\_NOT\_FOUND, MC\_SUCCESS, RWLOCK\_WRLock, RWLOCK\_WRunLock, barrier\_queue\_s::size, and list\_s::size.

Referenced by MC\_BarrierDelete().

#### 13.24.2.5 int barrier\_queue\_Destroy (barrier\_queue\_p queue)

Definition at line 131 of file barrier.c.

References barrier\_node\_Destroy(), barrier\_queue\_Pop(), barrier\_queue\_s::list, ListTerminate(), barrier\_queue\_s::lock, MC\_SUCCESS, node, and RWLOCK\_DESTROY.

Referenced by mc\_platform\_Destroy().

#### 13.24.2.6 barrier\_node\_p barrier\_queue\_Get (barrier\_queue\_p list, int id)

Definition at line 145 of file barrier.c.

References `barrier_queue_s::list`, `list_s::listhead`, `barrier_queue_s::lock`, `listNode_s::next`, `listNode_s::node_data`, `RWLOCK_RDLOCK`, and `RWLOCK_RDUNLOCK`.

Referenced by `MC_Barrier()`, and `MC_BarrierInit()`.

#### **13.24.2.7 `barrier_queue_p barrier_queue_New (void)`**

Definition at line 162 of file `barrier.c`.

References `CHECK_NULL`, `barrier_queue_s::list`, `ListInitialize()`, `barrier_queue_s::lock`, `RWLOCK_INIT`, and `RWLOCK_T`.

Referenced by `mc_platform_Initialize()`.

#### **13.24.2.8 `barrier_node_p barrier_queue_Pop (barrier_queue_p queue)`**

Definition at line 176 of file `barrier.c`.

References `barrier_queue_s::list`, `ListPop()`, and `node`.

Referenced by `barrier_queue_Destroy()`.

## 13.25 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/cmd\_prompt.h File Reference

```
#include "macros.h"
```

### Data Structures

- struct [cmd\\_prompt\\_s](#)
- struct [command\\_s](#)

### Defines

- #define [CMDLINE\\_SIZE](#) 80

### Typedefs

- typedef struct [cmd\\_prompt\\_s](#) [cmd\\_prompt\\_t](#)
- typedef [cmd\\_prompt\\_t](#) \* [cmd\\_prompt\\_p](#)
- typedef struct [command\\_s](#) [command\\_t](#)
- typedef [command\\_t](#) \* [command\\_p](#)

### Functions

- [int cmd\\_prompt\\_Destroy](#) ([cmd\\_prompt\\_p](#) cmd\_prompt)
- [cmd\\_prompt\\_p cmd\\_prompt\\_Initialize](#) (struct [mc\\_platform\\_s](#) \*[mc\\_platform](#))
- [void cmd\\_prompt\\_Start](#) (struct [mc\\_platform\\_s](#) \*[mc\\_platform](#))
- [int split\\_string](#) (char \*\*\*args, const char \*buf)
- [int process\\_command](#) ([command\\_t](#) \*cmd)
- [void \\* cmd\\_prompt\\_Thread](#) (void \*arg)
- [int exec\\_command](#) ([command\\_t](#) cmd, struct [mc\\_platform\\_s](#) \*global)
- [int dealloc\\_command](#) ([command\\_t](#) \*cmd)

#### 13.25.1 Define Documentation

##### 13.25.1.1 #define CMDLINE\_SIZE 80

Definition at line 37 of file cmd\_prompt.h.

#### 13.25.2 Typedef Documentation

##### 13.25.2.1 typedef cmd\_prompt\_t\* cmd\_prompt\_p

Definition at line 44 of file cmd\_prompt.h.

**13.25.2.2** `typedef struct cmd_prompt_s cmd_prompt_t`

**13.25.2.3** `typedef command_t* command_p`

Definition at line 51 of file cmd\_prompt.h.

**13.25.2.4** `typedef struct command_s command_t`

### **13.25.3 Function Documentation**

**13.25.3.1** `int cmd_prompt_Destroy (cmd_prompt_p cmd_prompt)`

Definition at line 129 of file cmd\_prompt.c.

References MC\_SUCCESS.

Referenced by mc\_platform\_Destroy().

**13.25.3.2** `cmd_prompt_p cmd_prompt_Initialize (struct mc_platform_s * mc_platform)`

**13.25.3.3** `void cmd_prompt_Start (struct mc_platform_s * mc_platform)`

**13.25.3.4** `void* cmd_prompt_Thread (void * arg)`

Definition at line 168 of file cmd\_prompt.c.

References command\_s::args, buf, dealloc\_command(), exec\_command(), command\_s::index, mc\_platform, command\_s::num\_args, process\_command(), and split\_string().

Referenced by cmd\_prompt\_Start().

**13.25.3.5** `int dealloc_command (command_t * cmd)`

Definition at line 306 of file cmd\_prompt.c.

References command\_s::args, and command\_s::num\_args.

Referenced by cmd\_prompt\_Thread().

**13.25.3.6** `int exec_command (command_t cmd, struct mc_platform_s * global)`

**13.25.3.7** `int process_command (command_t * cmd)`

Definition at line 272 of file cmd\_prompt.c.

References command\_s::args, command\_cmds, command\_s::index, and command\_s::num\_args.

Referenced by cmd\_prompt\_Thread().

**13.25.3.8** `int split_string (char *** args, const char * buf)`

Definition at line 224 of file cmd\_prompt.c.

References int.

Referenced by `cmd_prompt_Thread()`.

## 13.26 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/commands.h File Reference

```
#include "macros.h"
#include "mc_platform.h"
#include "commands.x.h"
```

### Defines

- #define [COMMAND](#)(name, cmd, desc) [int](#) handler\_##name (void \*arg, [mc\\_platform\\_p](#) global);
- #define [COMMAND](#)(name, cmd, description) [COMMAND\\_](#)##name ,
- #define [COMMAND](#)(name, cmd, description) &handler\_##name ,
- #define [COMMAND](#)(name, cmd, description) description ,
- #define [COMMAND](#)(name, cmd, description) cmd ,

### Typedefs

- typedef [int](#)(\* [cmd\\_handler\\_t](#))(void \*, [mc\\_platform\\_p](#) global)

### Enumerations

- enum [command\\_indices\\_e](#)

### Functions

- [COMMAND](#) (QUIT,"quit","This command ends the MobileC application.") [COMMAND](#)(HELP
- This command displays [help](#) for any command Usage This command sends a client to a host Usage This command prints the list entries in the connectList [COMMAND](#) (PRINTLIST\_MESSAGE,"printlist\_message","This command prints all the list entries on the linked list:\n \ message\_queue") [COMMAND](#)(PRINTLIST\_AGENTS
- This command displays [help](#) for any command Usage This command sends a client to a host Usage This command prints the list entries in the connectList This command prints all the agents on the system which are still n on the agent list n [COMMAND](#) (FLUSH\_AGENTS,"flush\_agents","This command flushes all of the agents on the system which are still\n\ on the agent list.\n") [COMMAND\\_COUNT](#)

### Variables

- [cmd\\_handler\\_t](#) cmd\_handlers []
- char \* [command\\_descriptions](#) []
- char \* [command\\_cmds](#) []

#### 13.26.1 Define Documentation

##### 13.26.1.1 #define [COMMAND](#)(name, cmd, description) cmd ,

Definition at line 48 of file commands.h.

**13.26.1.2 #define COMMAND(name, cmd, description) description ,**

Definition at line 48 of file commands.h.

**13.26.1.3 #define COMMAND(name, cmd, description) &handler\_##name ,**

Definition at line 48 of file commands.h.

**13.26.1.4 #define COMMAND(name, cmd, description) COMMAND\_##name ,**

Definition at line 48 of file commands.h.

**13.26.1.5 #define COMMAND(name, cmd, desc) int handler\_##name (void \*arg, mc\_platform\_p global);**

Definition at line 48 of file commands.h.

**13.26.2 Typedef Documentation****13.26.2.1 typedef int(\* cmd\_handler\_t)(void \*, mc\_platform\_p global)**

Definition at line 45 of file commands.h.

**13.26.3 Enumeration Type Documentation****13.26.3.1 enum command\_indices\_e**

Definition at line 47 of file commands.h.

**13.26.4 Function Documentation****13.26.4.1 This command displays help for any command Usage This command sends a client to a host Usage This command prints the list entries in the connectList This command prints all the agents on the system which are still n on the agent list n command\_indices\_e::COMMAND (FLUSH\_AGENTS, "flush\_agents", "This command flushes all of the agents on the system which are still\n\n on the agent list.\n")****Type Constraints****13.26.4.2 This command displays help for any command Usage This command sends a client to a host Usage This command prints the list entries in the connectList command\_indices\_e::COMMAND (PRINTLIST\_MESSAGE, "printlist\_message", "This command prints all the list entries on the linked list:\n\n message\_queue")****Type Constraints**



**13.26.4.3** `command_indices_e::COMMAND (QUIT, "quit", "This command ends the MobileC application.")`

### **13.26.5 Variable Documentation**

**13.26.5.1** `cmd_handler_t cmd_handlers[ ]`

Definition at line 54 of file commands.h.

Referenced by `exec_command()`.

**13.26.5.2** `char* command_cmds[ ]`

Definition at line 68 of file commands.h.

Referenced by `handler_HELP()`, and `process_command()`.

**13.26.5.3** `char* command_descriptions[ ]`

Definition at line 61 of file commands.h.

Referenced by `handler_HELP()`, and `handler_SEND()`.

## 13.27 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/commands.x.h File Reference

### Functions

- [COMMAND](#) (QUIT,"quit","This command ends the MobileC application.") [COMMAND](#)(HELP
- This command displays [help](#) for any command Usage This command sends a client to a host Usage This command prints the list entries in the connectList [COMMAND](#) (PRINTLIST\_MESSAGE,"printlist\_message","This command prints all the list entries on the linked list:\n \ message\_queue") [COMMAND](#)(PRINTLIST\_AGENTS

### Variables

- [help](#)
- This command displays [help](#) for any command Usage [is](#)
- This command displays [help](#) for any command Usage [send](#)
- This command displays [help](#) for any command Usage This command sends a client to a host Usage [print\\_connectlist](#)
- This command displays [help](#) for any command Usage This command sends a client to a host Usage This command prints the list entries in the connectList [printlist\\_agents](#)

### 13.27.1 Function Documentation

- 13.27.1.1 This command displays help for any command Usage This command sends a client to a host Usage This command prints the list entries in the connectList [COMMAND](#) (PRINTLIST\_MESSAGE, "printlist\_message", "This command prints all the list entries on the linked list:\n \ message\_queue")**

#### Type Constraints

- 13.27.1.2 [COMMAND](#) (QUIT, "quit", "This command ends the MobileC application.")**

### 13.27.2 Variable Documentation

#### 13.27.2.1 [help](#)

Definition at line 40 of file commands.x.h.

- 13.27.2.2 This command displays help for any command Usage This command sends a client to a host Usage [is](#)**

Definition at line 40 of file commands.x.h.

- 13.27.2.3 This command displays help for any command Usage This command sends a client to a host Usage [print\\_connectlist](#)**

Definition at line 40 of file commands.x.h.

**13.27.2.4 This command displays help for any command Usage This command sends a client to a host Usage This command prints the list entries in the connectList printlist\_agents**

Definition at line 55 of file commands.x.h.

**13.27.2.5 This command displays help for any command Usage send**

Definition at line 40 of file commands.x.h.

Referenced by `extract_nonce_from_MA()`, `initiate_migration_process()`, `message_send_Thread()`, `mtp_http_InitializeFromConnection()`, `reply_migration_process()`, and `send_AES_en_MA()`.

## 13.28 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/connection.h File Reference

```
#include <sys/types.h>
#include <netinet/in.h>
```

### Data Structures

- struct [connection\\_s](#)

### Typedefs

- typedef struct [connection\\_s](#) [connection\\_t](#)
- typedef [connection\\_t](#) \* [connection\\_p](#)

### Functions

- [connection\\_p](#) [connection\\_New](#) (void)
- [int](#) [connection\\_Destroy](#) ([connection\\_p](#) connection)
- [connection\\_p](#) [connection\\_Copy](#) ([connection\\_p](#) connection)
- void [connection\\_Close](#) ([connection\\_p](#) connection)
- void [connection\\_SetHosts](#) ([connection\\_p](#) connection, char \*remote\_host, char \*local\_host)
- [int](#) [connection\\_CreateMessage](#) ([connection\\_p](#) connection)
- [int](#) [connection\\_SendtoConnection](#) ([connection\\_p](#) connection, char \*send\_buffer)
- [int](#) [connection\\_RecvFromConnection](#) ([connection\\_p](#) connection, struct [mc\\_platform\\_s](#) \*global)

### 13.28.1 Typedef Documentation

#### 13.28.1.1 typedef [connection\\_t](#)\* [connection\\_p](#)

Definition at line 66 of file [connection.h](#).

#### 13.28.1.2 typedef struct [connection\\_s](#) [connection\\_t](#)

### 13.28.2 Function Documentation

#### 13.28.2.1 void [connection\\_Close](#) ([connection\\_p](#) *connection*)

#### 13.28.2.2 [connection\\_p](#) [connection\\_Copy](#) ([connection\\_p](#) *connection*)

Definition at line 80 of file [connection.c](#).

References [connection\\_s::addr](#), [connection\\_s::clientfd](#), [connection\\_s::connect\\_id](#), [connection\\_New\(\)](#), [connection\\_s::remote\\_hostname](#), and [connection\\_s::serverfd](#).

**13.28.2.3 int connection\_CreateMessage (connection\_p *connection*)**

**13.28.2.4 int connection\_Destroy (connection\_p *connection*)**

Definition at line 48 of file connection.c.

References connection\_s::clientfd, MC\_SUCCESS, and connection\_s::remote\_hostname.

Referenced by acc\_connection\_Thread().

**13.28.2.5 connection\_p connection\_New (void)**

Definition at line 69 of file connection.c.

Referenced by connection\_Copy(), and listen\_Thread().

**13.28.2.6 int connection\_RecvFromConnection (connection\_p *connection*, struct mc\_platform\_s \*  
*global*)**

**13.28.2.7 int connection\_SendtoConnection (connection\_p *connection*, char \* *send\_buffer*)**

**13.28.2.8 void connection\_SetHosts (connection\_p *connection*, char \* *remote\_host*, char \*  
*local\_host*)**

## 13.29 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/data\_structures.h File Reference

```
#include "ap_queue_template.h"
#include "../mc_list/list.h"
#include "connection.h"
#include "fipa_acl.h"
#include "agent_mailbox.h"
#include "message.h"
#include "interpreter_variable_data.h"
```

### Functions

- [AP\\_QUEUE\\_DECL\\_TEMPLATE](#) (connection\_queue, connection) [AP\\_QUEUE\\_GENERIC\\_DECL\\_TEMPLATE](#)(connection\_queue
- [connection\\_t](#) [int](#) [AP\\_QUEUE\\_GENERIC\\_DECL\\_TEMPLATE](#) (connection\_queue, [Remove](#), [int](#), [int](#)) [int](#) [connection\\_queue\\_Print](#)(connection\_queue\_p clist)
- [AP\\_QUEUE\\_DECL\\_TEMPLATE](#) (message\_queue, message) [int](#) [message\\_queue\\_Print](#)(message\_queue\_p queue)
- [AP\\_QUEUE\\_DECL\\_TEMPLATE](#) (agent\_queue, agent) [AP\\_QUEUE\\_GENERIC\\_DECL\\_TEMPLATE](#)(agent\_queue
- [struct agent\\_s](#) [int](#) [int](#) [AP\\_QUEUE\\_GENERIC\\_DECL\\_TEMPLATE](#) (agent\_queue, [SearchName](#), [struct agent\\_s](#) \*, [char](#) \*) [AP\\_QUEUE\\_GENERIC\\_DECL\\_TEMPLATE](#)(agent\_queue
- [struct agent\\_s](#) [int](#) [int](#) [AP\\_QUEUE\\_GENERIC\\_DECL\\_TEMPLATE](#) (agent\_queue, [RemoveName](#), [int](#), [char](#) \*) [int](#) [agent\\_queue\\_Print](#)(agent\_queue\_p queue)
- [AP\\_QUEUE\\_DECL\\_TEMPLATE](#) (agent\_variable\_list, interpreter\_variable\_data) [AP\\_QUEUE\\_GENERIC\\_DECL\\_TEMPLATE](#)(agent\_variable\_list
- [interpreter\\_variable\\_data\\_t](#) [char](#) \* [AP\\_QUEUE\\_GENERIC\\_DECL\\_TEMPLATE](#) (agent\_variable\_list, [Remove](#), [int](#), [char](#) \*) [AP\\_QUEUE\\_DECL\\_TEMPLATE](#)(mail\_queue
- [interpreter\\_variable\\_data\\_t](#) [char](#) \*fipa\_acl\_message [AP\\_QUEUE\\_GENERIC\\_DECL\\_TEMPLATE](#) (mail\_queue, [SearchReceivers](#), [fipa\\_acl\\_message\\_p](#), [char](#) \*) [AP\\_QUEUE\\_GENERIC\\_DECL\\_TEMPLATE](#)(mail\_queue
- [interpreter\\_variable\\_data\\_t](#) [char](#) \*fipa\_acl\_message [char](#) \* [AP\\_QUEUE\\_DECL\\_TEMPLATE](#) (mailbox\_queue, agent\_mailbox) [AP\\_QUEUE\\_GENERIC\\_DECL\\_TEMPLATE](#)(mailbox\_queue
- [interpreter\\_variable\\_data\\_t](#) [char](#) \*fipa\_acl\_message [char](#) [char](#) \* [AP\\_QUEUE\\_GENERIC\\_DECL\\_TEMPLATE](#) (mailbox\_queue, [SearchID](#), [agent\\_mailbox\\_p](#), [int](#)) [AP\\_QUEUE\\_GENERIC\\_DECL\\_TEMPLATE](#)(mailbox\_queue

### Variables

- [Search](#)
- [struct agent\\_s](#) [int](#) [Remove](#)
- [struct agent\\_s](#) [int](#) [int](#)
- [interpreter\\_variable\\_data\\_t](#) [char](#) \*fipa\_acl\_message [RemoveName](#)
- [interpreter\\_variable\\_data\\_t](#) [char](#) \*fipa\_acl\_message [char](#) \* [SearchName](#)
- [interpreter\\_variable\\_data\\_t](#) [char](#) \*fipa\_acl\_message [char](#) [agent\\_mailbox\\_p](#)
- [interpreter\\_variable\\_data\\_t](#) [char](#) \*fipa\_acl\_message [char](#) [char](#) \* [RemoveID](#)

## 13.29.1 Function Documentation

13.29.1.1 `interpreter_variable_data_t char* fipa_acl_message char*`  
`AP_QUEUE_DECL_TEMPLATE (mailbox_queue, agent_mailbox)`

### Type Constraints

13.29.1.2 `AP_QUEUE_DECL_TEMPLATE (agent_variable_list, interpreter_variable_data)`

13.29.1.3 `AP_QUEUE_DECL_TEMPLATE (agent_queue, agent)`

13.29.1.4 `AP_QUEUE_DECL_TEMPLATE (message_queue, message)`

13.29.1.5 `AP_QUEUE_DECL_TEMPLATE (connection_queue, connection)`

13.29.1.6 `interpreter_variable_data_t char* fipa_acl_message char char*`  
`AP_QUEUE_GENERIC_DECL_TEMPLATE (mailbox_queue, SearchID,`  
`agent_mailbox_p, int)`

### Type Constraints

13.29.1.7 `interpreter_variable_data_t char* fipa_acl_message AP_QUEUE_GENERIC-`  
`DECL_TEMPLATE (mail_queue, SearchReceivers, fipa_acl_message_p, char`  
`*)`

13.29.1.8 `interpreter_variable_data_t char* AP_QUEUE_GENERIC_DECL_TEMPLATE`  
`(agent_variable_list, Remove, int, char *)`

### Type Constraints

13.29.1.9 `struct agent_s int int AP_QUEUE_GENERIC_DECL_TEMPLATE (agent_queue,`  
`RemoveName, int, char *)`

### Type Constraints

13.29.1.10 `struct agent_s int AP_QUEUE_GENERIC_DECL_TEMPLATE (agent_queue,`  
`SearchName, struct agent_s *, char *)` **[read]**

### Type Constraints

**13.29.1.11** `connection_t` `int` `AP_QUEUE_GENERIC_DECL_TEMPLATE` (`connection_queue`,  
`Remove`, `int`, `int`)

#### Type Constraints

### 13.29.2 Variable Documentation

**13.29.2.1** `interpreter_variable_data_t` `char*` `fipa_acl_message` `char` `agent_mailbox_p`

Definition at line 162 of file `data_structures.h`.

**13.29.2.2** `interpreter_variable_data_t` `char` `*fipa_acl_message` `char` `char` `int`

Definition at line 101 of file `data_structures.h`.

Referenced by `aes_en_de()`, `LibMC::MCAgency::ChInitializeOptions()`, `default_callback()`, `main()`, `mpi_lsb()`, `mpi_read_string()`, `net_accept()`, `rece_de_msg()`, `remove_nonce_from_MA()`, `rsa_pkcs1_decrypt()`, `rsa_pkcs1_verify()`, `send_AES_en_MA()`, `sha4_finish()`, `sha4_update()`, `split_string()`, and `ssl_parse_certificate()`.

**13.29.2.3** `struct` `agent_s` `int` `Remove`

Definition at line 101 of file `data_structures.h`.

**13.29.2.4** `interpreter_variable_data_t` `char*` `fipa_acl_message` `char` `char*` `RemoveID`

Definition at line 176 of file `data_structures.h`.

**13.29.2.5** `interpreter_variable_data_t` `char*` `fipa_acl_message` `RemoveName`

Definition at line 149 of file `data_structures.h`.

**13.29.2.6** `Search`

Definition at line 53 of file `data_structures.h`.

**13.29.2.7** `interpreter_variable_data_t` `char*` `fipa_acl_message` `char*` `SearchName`

Definition at line 162 of file `data_structures.h`.



## 13.30 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/df.h File Reference

```
#include "../mc_list/list.h"
#include "../include/macros.h"
#include "df_request.x.h"
```

### Defines

- `#define REQUEST(name, string, description) REQUEST_##name,`
- `#define REQUEST(name, string, description) int request_handler_##name (struct mc_platform_s *global, void* data);`

### Enumerations

- enum `df_request_list_index_e`
- enum `service_types_e` {  
    `ZERO`, `MISC`, `INSERT`, `SOME`,  
    `TYPES`, `HERE` }

### Functions

- `REQUEST (SUBSCRIBE,"subscribe","Subscribe to a directory facilitator") REQUEST(REGISTER`
- Register services with the directory facilitator `REQUEST (DEREGISTER,"deregister","Deregisters mobile agent services from a directory facilitator.") REQUEST(SEARCH`
- `STRUCT (df_request_list_node, MUTEX_T *lock;COND_T *cond;const char *command;void *data;int data_size;) STRUCT(df_request_list`
- `STRUCT (df_search_results, char **agent_names;char **service_names;int *agent_ids;int num_results;) STRUCT(df_request_search`
- `STRUCT (df_node, MUTEX_T *lock;int agent_id;char *agent_name;int num_services;char **service_names;enum service_types_e service_types;) STRUCT(df`
- `STRUCT (df_deregister, int agent_id;char *service_name;) int df_Add(struct df_s *df`
- `int df_AddRequest (struct df_s *df, struct df_request_list_node_s *node)`
- `int df_Destroy (df_p df)`
- `df_p df_Initialize (struct mc_platform_s *mc_platform)`
- `int df_ProcessRequest (struct mc_platform_s *global)`
- `int df_SearchForService (df_p df, const char *searchstring, char ***agent_names, char ***service_names, int **agent_ids, int *num_entries)`
- `void df_Start (struct mc_platform_s *mc_platform)`
- `int df_node_Destroy (df_node_p df_node)`
- `int df_request_list_node_Destroy (df_request_list_node_p node)`
- `df_request_list_node_p df_request_list_node_New (void)`
- `int df_request_list_Destroy (df_request_list_p df_request_list)`
- `df_request_list_p df_request_list_New (void)`
- `df_request_list_node_p df_request_list_Pop (df_request_list_p requests)`
- `df_request_search_p df_request_search_New (void)`
- `int df_request_search_Destroy (df_request_search_p node)`
- `void * df_Thread (void *arg)`

## Variables

- `MUTEX_T * lock`
- `COND_T * cond`
- `int size`
- `list_p request_list`
- `char * search_string`
- `df_search_results_p search_results`
- `struct mc_platform_s * mc_platform`
- `list_p service_list`
- `int num_entries`
- `int waiting`
- `MUTEX_T * waiting_lock`
- `COND_T * waiting_cond`
- `THREAD_T thread`
- `struct df_node_s * node`

### 13.30.1 Define Documentation

#### 13.30.1.1 `#define REQUEST(name, string, description) int request_handler_##name (struct mc_platform_s *global, void* data);`

Definition at line 193 of file df.h.

#### 13.30.1.2 `#define REQUEST(name, string, description) REQUEST_##name,`

Definition at line 193 of file df.h.

### 13.30.2 Enumeration Type Documentation

#### 13.30.2.1 `enum df_request_list_index_e`

Definition at line 46 of file df.h.

#### 13.30.2.2 `enum service_types_e`

Enumerator:

***ZERO***  
***MISC***  
***INSERT***  
***SOME***  
***TYPES***  
***HERE***

Definition at line 53 of file df.h.

### 13.30.3 Function Documentation

#### 13.30.3.1 `int df_AddRequest (struct df_s * df, struct df_request_list_node_s * node)`

Definition at line 66 of file df.c.

References ListAdd(), and SIGNAL.

Referenced by MC\_DeregisterService(), MC\_RegisterService(), and MC\_SearchForService().

#### 13.30.3.2 `int df_Destroy (df_p df)`

Definition at line 83 of file df.c.

References COND\_DESTROY, df\_node\_Destroy(), df\_request\_list\_Destroy(), ListPop(), ListTerminate(), MC\_SUCCESS, MUTEX\_DESTROY, and MUTEX\_LOCK.

Referenced by mc\_platform\_Destroy().

#### 13.30.3.3 `df_p df_Initialize (struct mc_platform_s * mc_platform)`

#### 13.30.3.4 `int df_node_Destroy (df_node_p df_node)`

Definition at line 412 of file df.c.

References MC\_SUCCESS, and MUTEX\_LOCK.

Referenced by df\_Destroy().

#### 13.30.3.5 `int df_ProcessRequest (struct mc_platform_s * global)`

Definition at line 132 of file df.c.

References mc\_platform\_s::df, df\_request\_list\_Pop(), MC\_ERR\_EMPTY, and MC\_ERR\_INVALID.

Referenced by df\_Thread().

#### 13.30.3.6 `int df_request_list_Destroy (df_request_list_p df_request_list)`

Definition at line 321 of file df.c.

References df\_request\_list\_node\_Destroy(), ListPop(), ListTerminate(), MC\_SUCCESS, and node.

Referenced by df\_Destroy().

#### 13.30.3.7 `df_request_list_p df_request_list_New (void)`

Definition at line 343 of file df.c.

References CHECK\_NULL, COND\_INIT, COND\_T, ListInitialize(), MUTEX\_INIT, and MUTEX\_T.

Referenced by df\_Initialize().

#### 13.30.3.8 `int df_request_list_node_Destroy (df_request_list_node_p node)`

Definition at line 292 of file df.c.

References COND\_DESTROY, MC\_SUCCESS, and MUTEX\_DESTROY.

Referenced by df\_request\_list\_Destroy(), and MC\_SearchForService().

### 13.30.3.9 df\_request\_list\_node\_p df\_request\_list\_node\_New (void)

Definition at line 303 of file df.c.

References CHECK\_NULL, COND\_INIT, COND\_T, MUTEX\_INIT, MUTEX\_T, and node.

Referenced by MC\_DeregisterService(), MC\_RegisterService(), and MC\_SearchForService().

### 13.30.3.10 df\_request\_list\_node\_p df\_request\_list\_Pop (df\_request\_list\_p requests)

Definition at line 368 of file df.c.

References ListPop(), MUTEX\_LOCK, MUTEX\_UNLOCK, and node.

Referenced by df\_ProcessRequest().

### 13.30.3.11 int df\_request\_search\_Destroy (df\_request\_search\_p node)

Definition at line 399 of file df.c.

References COND\_DESTROY, MC\_SUCCESS, and MUTEX\_DESTROY.

Referenced by MC\_SearchForService().

### 13.30.3.12 df\_request\_search\_p df\_request\_search\_New (void)

Definition at line 384 of file df.c.

References CHECK\_NULL, COND\_INIT, COND\_T, MUTEX\_INIT, MUTEX\_T, and search.

Referenced by MC\_SearchForService().

### 13.30.3.13 int df\_SearchForService (df\_p df, const char \* searchstring, char \*\*\* agent\_names, char \*\*\* service\_names, int \*\* agent\_ids, int \* num\_entries)

Definition at line 176 of file df.c.

References MC\_ERR\_NOT\_FOUND, MC\_SUCCESS, MUTEX\_LOCK, MUTEX\_UNLOCK, listNode\_s::next, and listNode\_s::node\_data.

Referenced by request\_handler\_SEARCH().

### 13.30.3.14 void df\_Start (struct mc\_platform\_s \* mc\_platform)

### 13.30.3.15 void\* df\_Thread (void \* arg)

Definition at line 426 of file df.c.

References COND\_BROADCAST, COND\_WAIT, mc\_platform\_s::df, df\_ProcessRequest(), MC\_SUCCESS, MUTEX\_LOCK, MUTEX\_UNLOCK, mc\_platform\_s::quit, mc\_platform\_s::quit\_lock, and THREAD\_EXIT.

Referenced by df\_Start().

**13.30.3.16** Register services with the directory facilitator df\_request\_list\_index\_e::REQUEST (DEREGISTER, "deregister", "Deregisters mobile agent services from a directory facilitator.")

#### Type Constraints

**13.30.3.17** df\_request\_list\_index\_e::REQUEST (SUBSCRIBE, "subscribe", "Subscribe to a directory facilitator")

**13.30.3.18** STRUCT (df\_deregister, int agent\_id;char \*service\_name;)

**13.30.3.19** STRUCT (df\_node, MUTEX\_T \*lock;int agent\_id;char \*agent\_name;int num\_services;char \*\*service\_names;enum service\_types\_e service\_types;)

**13.30.3.20** STRUCT (df\_search\_results, char \*\*agent\_names;char \*\*service\_names;int \*agent\_ids;int num\_results;)

**13.30.3.21** STRUCT (df\_request\_list\_node, MUTEX\_T \*lock;COND\_T \*cond;const char \*command;void \*data;int data\_size;)

### 13.30.4 Variable Documentation

#### 13.30.4.1 COND\_T \* cond

Definition at line 74 of file df.h.

#### 13.30.4.2 MUTEX\_T \* lock

Definition at line 73 of file df.h.

#### 13.30.4.3 struct mc\_platform\_s\* mc\_platform

Definition at line 107 of file df.h.

Referenced by acc\_connection\_Thread(), acc\_MessageHandlerThread(), acc\_Thread(), agent\_RunChScriptThread(), ams\_Thread(), cmd\_prompt\_Thread(), listen\_Thread(), MC\_Initialize(), mc\_platform\_Initialize(), MC\_WaitAgent(), message\_send\_Thread(), and udplisten\_Thread().

#### 13.30.4.4 struct df\_node\_s\* node

Definition at line 129 of file df.h.

Referenced by add\_variable(), agent\_queue\_Flush(), agent\_xml\_compose(), agent\_xml\_compose\_\_agent\_code(), agent\_xml\_compose\_\_agent\_data(), agent\_xml\_compose\_\_create\_row\_nodes(), agent\_xml\_compose\_\_data(), agent\_xml\_compose\_\_gaf\_message(), agent\_xml\_compose\_\_home(), agent\_xml\_compose\_\_message(), agent\_xml\_compose\_\_mobile\_agent(), agent\_xml\_compose\_\_name(),

agent\_xml\_compose\_\_owner(), agent\_xml\_compose\_\_row(), agent\_xml\_compose\_\_task(), agent\_xml\_compose\_\_tasks(), agent\_xml\_compose\_\_wg\_code(), AP\_QUEUE\_SEARCH\_TEMPLATE(), AP\_QUEUE\_STD\_DEFN\_TEMPLATE(), barrier\_node\_Initialize(), barrier\_queue\_Destroy(), barrier\_queue\_Pop(), df\_request\_list\_Destroy(), df\_request\_list\_node\_New(), df\_request\_list\_Pop(), fipa\_envelope\_Compose(), fipa\_envelope\_Compose\_\_acl\_representation(), fipa\_envelope\_Compose\_\_date(), fipa\_envelope\_Compose\_\_envelope(), fipa\_envelope\_Compose\_\_intended\_receiver(), fipa\_envelope\_Compose\_\_params(), fipa\_envelope\_Compose\_\_payload\_encoding(), fipa\_envelope\_Compose\_\_to(), fipa\_envelope\_HandleReceived(), main(), MC\_Barrier(), MC\_BarrierInit(), MC\_SyncInit(), mxml\_new(), mxmlNewCustom(), mxmlNewElement(), mxmlNewInteger(), mxmlNewOpaque(), mxmlNewReal(), mxmlNewText(), mxmlNewTextf(), request\_handler\_DEREGISTER(), scan\_file(), syncListNew(), write\_element(), xml\_get\_deep\_child(), and xml\_new\_cdata().

#### 13.30.4.5 int num\_entries

Definition at line 113 of file df.h.

#### 13.30.4.6 df\_request\_list\_p request\_list

Definition at line 76 of file df.h.

#### 13.30.4.7 df\_search\_results\_p search\_results

Definition at line 91 of file df.h.

#### 13.30.4.8 char\* search\_string

Definition at line 90 of file df.h.

#### 13.30.4.9 list\_p service\_list

Definition at line 110 of file df.h.

#### 13.30.4.10 int size

Definition at line 75 of file df.h.

Referenced by agent\_AddPersistentVariable(), agent\_xml\_compose\_\_create\_row\_nodes(), interpreter\_variable\_data\_Initialize(), interpreter\_variable\_data\_InitializeFromAgent(), MC\_GetAgentReturnData(), MC\_SaveData\_chdl(), MC\_WaitAgent(), mxml\_vsnprintf(), receive\_AES\_en\_MA(), and send\_AES\_en\_MA().

#### 13.30.4.11 THREAD\_T thread

Definition at line 117 of file df.h.

#### 13.30.4.12 int waiting

Definition at line 114 of file df.h.

**13.30.4.13 COND\_T\* waiting\_cond**

Definition at line 116 of file df.h.

**13.30.4.14 MUTEX\_T\* waiting\_lock**

Definition at line 115 of file df.h.

## 13.31 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/df\_request.x.h File Reference

### Functions

- [REQUEST](#) (SUBSCRIBE,"subscribe","Subscribe to a directory facilitator") [REQUEST](#)(REGISTER
- Register services with the directory facilitator [REQUEST](#) (DEREGISTER,"deregister","Deregisters mobile agent services from a directory facilitator.") [REQUEST](#)(SEARCH

### Variables

- [register](#)
- Register services with the directory facilitator [search](#)

### 13.31.1 Function Documentation

**13.31.1.1 Register services with the directory facilitator [REQUEST](#) (DEREGISTER, "deregister", "Deregisters mobile agent services from a directory facilitator.")**

#### Type Constraints

**13.31.1.2 [REQUEST](#) (SUBSCRIBE, "subscribe", "Subscribe to a directory facilitator")**

### 13.31.2 Variable Documentation

#### 13.31.2.1 [register](#)

Definition at line 43 of file df\_request.x.h.

#### 13.31.2.2 Register services with the directory facilitator [search](#)

Definition at line 54 of file df\_request.x.h.

Referenced by [df\\_request\\_search\\_New\(\)](#), [MC\\_SearchForService\(\)](#), and [request\\_handler\\_SEARCH\(\)](#).



## 13.32 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/dynstring.h File Reference

### Data Structures

- struct [dynstring\\_s](#)

### Defines

- #define [COMPOSE\\_BLOCKSIZE](#) 1024

### Typedefs

- typedef struct [dynstring\\_s](#) [dynstring\\_t](#)
- typedef [dynstring\\_t](#) \* [dynstring\\_p](#)

### Functions

- [dynstring\\_t](#) \* [dynstring\\_New](#) (void)
- int [dynstring\\_Append](#) ([dynstring\\_t](#) \*msg, char \*str)
- int [dynstring\\_Destroy](#) ([dynstring\\_t](#) \*dynstring)

#### 13.32.1 Define Documentation

##### 13.32.1.1 #define COMPOSE\_BLOCKSIZE 1024

Definition at line 8 of file [dynstring.h](#).

Referenced by [dynstring\\_Append\(\)](#), and [dynstring\\_New\(\)](#).

#### 13.32.2 Typedef Documentation

##### 13.32.2.1 typedef [dynstring\\_t](#)\* [dynstring\\_p](#)

Definition at line 14 of file [dynstring.h](#).

##### 13.32.2.2 typedef struct [dynstring\\_s](#) [dynstring\\_t](#)

#### 13.32.3 Function Documentation

##### 13.32.3.1 int [dynstring\\_Append](#) ([dynstring\\_t](#) \*msg, char \*str)

Definition at line 30 of file [dynstring.c](#).

References [COMPOSE\\_BLOCKSIZE](#), [dynstring\\_s::len](#), [dynstring\\_s::message](#), and [dynstring\\_s::size](#).

Referenced by [fipa\\_acl\\_Compose\(\)](#), [fipa\\_agent\\_identifier\\_Compose\(\)](#), [fipa\\_agent\\_identifier\\_set\\_-Compose\(\)](#), [fipa\\_DateTime\\_Compose\(\)](#), [fipa\\_envelope\\_Compose\\_\\_from\(\)](#), [fipa\\_number\\_Compose\(\)](#),

fipa\_performative\_Compose(), fipa\_protocol\_Compose(), fipa\_string\_Compose(), fipa\_url\_Compose(), fipa\_url\_sequence\_Compose(), fipa\_word\_Compose(), message\_send\_Thread(), mtp\_http\_CreateMessage(), and mtp\_http\_InitializeFromConnection().

### 13.32.3.2 int dynstring\_Destroy (dynstring\_t \* *dynstring*)

Definition at line 56 of file dynstring.c.

References dynstring\_s::message.

Referenced by fipa\_envelope\_Compose\_\_from(), MC\_AclSend(), message\_send\_Thread(), mtp\_http\_CreateMessage(), and mtp\_http\_InitializeFromConnection().

### 13.32.3.3 dynstring\_t\* dynstring\_New (void)

Definition at line 14 of file dynstring.c.

References COMPOSE\_BLOCKSIZE, dynstring\_s::len, dynstring\_s::message, and dynstring\_s::size.

Referenced by fipa\_acl\_Compose(), fipa\_envelope\_Compose\_\_from(), message\_send\_Thread(), mtp\_http\_CreateMessage(), and mtp\_http\_InitializeFromConnection().

## 13.33 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/fipa\_acl.h File Reference

```
#include "dynstring.h"
```

### Data Structures

- struct [fipa\\_acl\\_message\\_s](#)
- struct [fipa\\_message\\_string\\_s](#)
- struct [fipa\\_url\\_sequence\\_s](#)
- struct [fipa\\_agent\\_identifier\\_set\\_s](#)
- struct [fipa\\_agent\\_identifier\\_s](#)
- struct [fipa\\_expression\\_s](#)
- union [fipa\\_expression\\_s::content\\_u](#)
- struct [fipa\\_word\\_s](#)
- struct [fipa\\_string\\_s](#)
- struct [fipa\\_DateTime\\_s](#)
- struct [fipa\\_url\\_s](#)
- struct [fipa\\_number\\_s](#)

### Typedefs

- typedef enum [fipa\\_expression\\_type\\_e](#) [fipa\\_expression\\_type\\_t](#)
- typedef struct [fipa\\_acl\\_message\\_s](#) [fipa\\_acl\\_message\\_t](#)
- typedef [fipa\\_acl\\_message\\_t](#) \* [fipa\\_acl\\_message\\_p](#)
- typedef struct [fipa\\_message\\_string\\_s](#) [fipa\\_message\\_string\\_t](#)
- typedef [fipa\\_message\\_string\\_t](#) \* [fipa\\_message\\_string\\_p](#)
- typedef struct [fipa\\_url\\_sequence\\_s](#) [fipa\\_url\\_sequence\\_t](#)
- typedef [fipa\\_url\\_sequence\\_t](#) \* [fipa\\_url\\_sequence\\_p](#)
- typedef struct [fipa\\_agent\\_identifier\\_set\\_s](#) [fipa\\_agent\\_identifier\\_set\\_t](#)
- typedef [fipa\\_agent\\_identifier\\_set\\_t](#) \* [fipa\\_agent\\_identifier\\_set\\_p](#)
- typedef struct [fipa\\_agent\\_identifier\\_s](#) [fipa\\_agent\\_identifier\\_t](#)
- typedef [fipa\\_agent\\_identifier\\_t](#) \* [fipa\\_agent\\_identifier\\_p](#)
- typedef struct [fipa\\_expression\\_s](#) [fipa\\_expression\\_t](#)
- typedef [fipa\\_expression\\_t](#) \* [fipa\\_expression\\_p](#)
- typedef struct [fipa\\_word\\_s](#) [fipa\\_word\\_t](#)
- typedef [fipa\\_word\\_t](#) \* [fipa\\_word\\_p](#)
- typedef struct [fipa\\_string\\_s](#) [fipa\\_string\\_t](#)
- typedef [fipa\\_string\\_t](#) \* [fipa\\_string\\_p](#)
- typedef struct [fipa\\_DateTime\\_s](#) [fipa\\_DateTime\\_t](#)
- typedef [fipa\\_DateTime\\_t](#) \* [fipa\\_DateTime\\_p](#)
- typedef struct [fipa\\_url\\_s](#) [fipa\\_url\\_t](#)
- typedef [fipa\\_url\\_t](#) \* [fipa\\_url\\_p](#)
- typedef struct [fipa\\_number\\_s](#) [fipa\\_number\\_t](#)
- typedef [fipa\\_number\\_t](#) \* [fipa\\_number\\_p](#)

## Enumerations

- enum `fipa_performative_e` {  
`FIPA_ERROR = -1, FIPA_ZERO, FIPA_ACCEPT_PROPOSAL, FIPA_AGREE,`  
`FIPA_CANCEL, FIPA_CALL_FOR_PROPOSAL, FIPA_CONFIRM, FIPA_DISCONFIRM,`  
`FIPA_FAILURE, FIPA_INFORM, FIPA_INFORM_IF, FIPA_INFORM_REF,`  
`FIPA_NOT_UNDERSTOOD, FIPA_PROPOGATE, FIPA_PROPOSE, FIPA_PROXY,`  
`FIPA_QUERY_IF, FIPA_QUERY_REF, FIPA_REFUSE, FIPA_REJECT_PROPOSAL,`  
`FIPA_REQUEST, FIPA_REQUEST_WHEN, FIPA_REQUEST_WHENEVER, FIPA_-`  
`SUBSCRIBE,`  
`FIPA_ERROR = -1, FIPA_ZERO, FIPA_ACCEPT_PROPOSAL, FIPA_AGREE,`  
`FIPA_CANCEL, FIPA_CALL_FOR_PROPOSAL, FIPA_CONFIRM, FIPA_DISCONFIRM,`  
`FIPA_FAILURE, FIPA_INFORM, FIPA_INFORM_IF, FIPA_INFORM_REF,`  
`FIPA_NOT_UNDERSTOOD, FIPA_PROPOGATE, FIPA_PROPOSE, FIPA_PROXY,`  
`FIPA_QUERY_IF, FIPA_QUERY_REF, FIPA_REFUSE, FIPA_REJECT_PROPOSAL,`  
`FIPA_REQUEST, FIPA_REQUEST_WHEN, FIPA_REQUEST_WHENEVER, FIPA_-`  
`SUBSCRIBE }`
- enum `fipa_protocol_e` {  
`FIPA_PROTOCOL_ERROR = -1, FIPA_PROTOCOL_NONE, FIPA_PROTOCOL_REQUEST,`  
`FIPA_PROTOCOL_QUERY,`  
`FIPA_PROTOCOL_REQUEST_WHEN, FIPA_PROTOCOL_CONTRACT_NET, FIPA_-`  
`PROTOCOL_ITERATED_CONTRACT_NET, FIPA_PROTOCOL_ENGLISH_AUCTION,`  
`FIPA_PROTOCOL_DUTCH_AUCTION, FIPA_PROTOCOL_BROKERING, FIPA_-`  
`PROTOCOL_RECRUITING, FIPA_PROTOCOL_SUBSCRIBE,`  
`FIPA_PROTOCOL_PROPOSE, FIPA_PROTOCOL_END }`
- enum `fipa_expression_type_e` {  
`FIPA_EXPR_ZERO, FIPA_EXPR_WORD, FIPA_EXPR_STRING, FIPA_EXPR_NUMBER,`  
`FIPA_EXPR_DATETIME, FIPA_EXPR_EXPRESSION }`

## Functions

- `int fipa_performative_Compose (dynstring_t *msg, enum fipa_performative_e performative)`
- `int fipa_protocol_Compose (dynstring_t *msg, enum fipa_protocol_e protocol)`
- `fipa_acl_message_t * fipa_acl_message_New (void)`
- `int fipa_acl_message_Destroy (fipa_acl_message_t *message)`
- `fipa_acl_message_t * fipa_acl_message_Copy (fipa_acl_message_t *src)`
- `int fipa_acl_Compose (dynstring_t **msg, fipa_acl_message_t *acl)`
- `fipa_message_string_t * fipa_message_string_New (void)`
- `int fipa_message_string_Destroy (fipa_message_string_t *message)`
- `fipa_message_string_t * fipa_message_string_Copy (fipa_message_string_t *src)`
- `fipa_url_sequence_t * fipa_url_sequence_New (void)`
- `int fipa_url_sequence_Destroy (fipa_url_sequence_t *sequence)`
- `fipa_url_sequence_t * fipa_url_sequence_Copy (fipa_url_sequence_t *src)`
- `int fipa_url_sequence_Compose (dynstring_t *msg, fipa_url_sequence_t *urls)`
- `fipa_agent_identifier_set_t * fipa_agent_identifier_set_New (void)`
- `int fipa_agent_identifier_set_Destroy (fipa_agent_identifier_set_t *idset)`

- [fipa\\_agent\\_identifier\\_set\\_t \\* fipa\\_agent\\_identifier\\_set\\_Copy \(fipa\\_agent\\_identifier\\_set\\_t \\*src\)](#)
- [int fipa\\_agent\\_identifier\\_set\\_Compose \(dynstring\\_t \\*msg, fipa\\_agent\\_identifier\\_set\\_t \\*ids\)](#)
- [fipa\\_agent\\_identifier\\_t \\* fipa\\_agent\\_identifier\\_New \(void\)](#)
- [int fipa\\_agent\\_identifier\\_Destroy \(fipa\\_agent\\_identifier\\_t \\*id\)](#)
- [fipa\\_agent\\_identifier\\_t \\* fipa\\_agent\\_identifier\\_Copy \(fipa\\_agent\\_identifier\\_t \\*src\)](#)
- [int fipa\\_agent\\_identifier\\_Compose \(dynstring\\_t \\*msg, fipa\\_agent\\_identifier\\_t \\*id\)](#)
- [fipa\\_expression\\_t \\* fipa\\_expression\\_New \(void\)](#)
- [int fipa\\_expression\\_Destroy \(fipa\\_expression\\_t \\*expr\)](#)
- [fipa\\_expression\\_t \\* fipa\\_expression\\_Copy \(fipa\\_expression\\_t \\*src\)](#)
- [int fipa\\_expression\\_Compose \(dynstring\\_t \\*msg, fipa\\_expression\\_t \\*expr\)](#)
- [fipa\\_word\\_t \\* fipa\\_word\\_New \(void\)](#)
- [int fipa\\_word\\_Destroy \(fipa\\_word\\_t \\*word\)](#)
- [fipa\\_word\\_t \\* fipa\\_word\\_Copy \(fipa\\_word\\_t \\*src\)](#)
- [int fipa\\_word\\_Compose \(dynstring\\_t \\*msg, fipa\\_word\\_t \\*word\)](#)
- [fipa\\_string\\_t \\* fipa\\_string\\_New \(void\)](#)
- [int fipa\\_string\\_Destroy \(fipa\\_string\\_t \\*str\)](#)
- [fipa\\_string\\_t \\* fipa\\_string\\_Copy \(fipa\\_string\\_t \\*src\)](#)
- [int fipa\\_string\\_Compose \(dynstring\\_t \\*msg, fipa\\_string\\_t \\*string\)](#)
- [fipa\\_DateTime\\_t \\* fipa\\_DateTime\\_New \(void\)](#)
- [int fipa\\_DateTime\\_Destroy \(fipa\\_DateTime\\_t \\*dt\)](#)
- [fipa\\_DateTime\\_t \\* fipa\\_DateTime\\_Copy \(fipa\\_DateTime\\_t \\*src\)](#)
- [int fipa\\_DateTime\\_Compose \(dynstring\\_t \\*msg, fipa\\_DateTime\\_t \\*date\)](#)
- [fipa\\_url\\_t \\* fipa\\_url\\_New \(void\)](#)
- [int fipa\\_url\\_Destroy \(fipa\\_url\\_t \\*url\)](#)
- [fipa\\_url\\_t \\* fipa\\_url\\_Copy \(fipa\\_url\\_t \\*src\)](#)
- [int fipa\\_url\\_Compose \(dynstring\\_t \\*msg, fipa\\_url\\_t \\*url\)](#)
- [fipa\\_number\\_t \\* fipa\\_number\\_New \(void\)](#)
- [int fipa\\_number\\_Destroy \(fipa\\_number\\_t \\*number\)](#)
- [fipa\\_number\\_t \\* fipa\\_number\\_Copy \(fipa\\_number\\_t \\*src\)](#)
- [int fipa\\_number\\_Compose \(dynstring\\_t \\*msg, fipa\\_number\\_t \\*number\)](#)
- [int fipa\\_acl\\_Parse \(struct fipa\\_acl\\_message\\_s \\*acl, fipa\\_message\\_string\\_p message\)](#)
- [int fipa\\_message\\_parameter\\_Parse \(struct fipa\\_acl\\_message\\_s \\*acl, fipa\\_message\\_string\\_p message\)](#)
- [int fipa\\_protocol\\_type\\_Parse \(enum fipa\\_protocol\\_e \\*protocol, fipa\\_message\\_string\\_p message\)](#)
- [int fipa\\_message\\_type\\_Parse \(enum fipa\\_performative\\_e \\*performative, fipa\\_message\\_string\\_p message\)](#)
- [int fipa\\_GetAtom \(fipa\\_message\\_string\\_p message, char expected\\_atom\)](#)
- [int fipa\\_word\\_Parse \(fipa\\_word\\_t \\*\\*word, fipa\\_message\\_string\\_p message\)](#)
- [int fipa\\_CheckNextToken \(const fipa\\_message\\_string\\_p message, const char \\*token\)](#)
- [int fipa\\_expression\\_Parse \(fipa\\_expression\\_t \\*\\*expression, fipa\\_message\\_string\\_p message\)](#)
- [int fipa\\_GetNextWord \(char \\*\\*word, const fipa\\_message\\_string\\_p message\)](#)
- [int fipa\\_GetWholeToken \(char \\*\\*word, const fipa\\_message\\_string\\_p message\)](#)
- [int fipa\\_datetime\\_Parse \(fipa\\_DateTime\\_p \\*datetime, fipa\\_message\\_string\\_p message\)](#)
- [int fipa\\_string\\_Parse \(fipa\\_string\\_p \\*fipa\\_string, fipa\\_message\\_string\\_p message\)](#)
- [int fipa\\_agent\\_identifier\\_Parse \(fipa\\_agent\\_identifier\\_p \\*aid, fipa\\_message\\_string\\_p message\)](#)
- [int fipa\\_agent\\_identifier\\_set\\_Parse \(fipa\\_agent\\_identifier\\_set\\_p \\*agent\\_ids, fipa\\_message\\_string\\_p message\)](#)
- [int fipa\\_url\\_sequence\\_Parse \(fipa\\_url\\_sequence\\_p \\*urls, fipa\\_message\\_string\\_p message\)](#)
- [int fipa\\_url\\_Parse \(fipa\\_url\\_p \\*url, fipa\\_message\\_string\\_p message\)](#)
- [struct fipa\\_acl\\_message\\_s \\* fipa\\_Reply \(struct fipa\\_acl\\_message\\_s \\*acl\)](#)

### 13.33.1 Typedef Documentation

#### 13.33.1.1 typedef fipa\_acl\_message\_t\* fipa\_acl\_message\_p

Definition at line 136 of file fipa\_acl.h.

#### 13.33.1.2 typedef struct fipa\_acl\_message\_s fipa\_acl\_message\_t

#### 13.33.1.3 typedef fipa\_agent\_identifier\_t\* fipa\_agent\_identifier\_p

Definition at line 186 of file fipa\_acl.h.

#### 13.33.1.4 typedef fipa\_agent\_identifier\_set\_t\* fipa\_agent\_identifier\_set\_p

Definition at line 172 of file fipa\_acl.h.

#### 13.33.1.5 typedef struct fipa\_agent\_identifier\_set\_s fipa\_agent\_identifier\_set\_t

#### 13.33.1.6 typedef struct fipa\_agent\_identifier\_s fipa\_agent\_identifier\_t

#### 13.33.1.7 typedef fipa\_DateTime\_t\* fipa\_DateTime\_p

Definition at line 246 of file fipa\_acl.h.

#### 13.33.1.8 typedef struct fipa\_DateTime\_s fipa\_DateTime\_t

#### 13.33.1.9 typedef fipa\_expression\_t\* fipa\_expression\_p

Definition at line 205 of file fipa\_acl.h.

#### 13.33.1.10 typedef struct fipa\_expression\_s fipa\_expression\_t

#### 13.33.1.11 typedef enum fipa\_expression\_type\_e fipa\_expression\_type\_t

#### 13.33.1.12 typedef fipa\_message\_string\_t\* fipa\_message\_string\_p

Definition at line 148 of file fipa\_acl.h.

#### 13.33.1.13 typedef struct fipa\_message\_string\_s fipa\_message\_string\_t

#### 13.33.1.14 typedef fipa\_number\_t\* fipa\_number\_p

Definition at line 268 of file fipa\_acl.h.

#### 13.33.1.15 typedef struct fipa\_number\_s fipa\_number\_t

#### 13.33.1.16 typedef fipa\_string\_t\* fipa\_string\_p

Definition at line 227 of file fipa\_acl.h.

**13.33.1.17**   `typedef struct fipa_string_s fipa_string_t`

**13.33.1.18**   `typedef fipa_url_t* fipa_url_p`

Definition at line 257 of file fipa\_acl.h.

**13.33.1.19**   `typedef fipa_url_sequence_t* fipa_url_sequence_p`

Definition at line 159 of file fipa\_acl.h.

**13.33.1.20**   `typedef struct fipa_url_sequence_s fipa_url_sequence_t`

**13.33.1.21**   `typedef struct fipa_url_s fipa_url_t`

**13.33.1.22**   `typedef fipa_word_t* fipa_word_p`

Definition at line 216 of file fipa\_acl.h.

**13.33.1.23**   `typedef struct fipa_word_s fipa_word_t`

## **13.33.2 Enumeration Type Documentation**

**13.33.2.1**   `enum fipa_expression_type_e`

Enumerator:

*FIPA\_EXPR\_ZERO*

*FIPA\_EXPR\_WORD*

*FIPA\_EXPR\_STRING*

*FIPA\_EXPR\_NUMBER*

*FIPA\_EXPR\_DATETIME*

*FIPA\_EXPR\_EXPRESSION*

Definition at line 94 of file fipa\_acl.h.

**13.33.2.2**   `enum fipa_performative_e`

Enumerator:

*FIPA\_ERROR*

*FIPA\_ZERO*

*FIPA\_ACCEPT\_PROPOSAL*

*FIPA\_AGREE*

*FIPA\_CANCEL*

*FIPA\_CALL\_FOR\_PROPOSAL*

*FIPA\_CONFIRM*

*FIPA\_DISCONFIRM*

*FIPA\_FAILURE*  
*FIPA\_INFORM*  
*FIPA\_INFORM\_IF*  
*FIPA\_INFORM\_REF*  
*FIPA\_NOT\_UNDERSTOOD*  
*FIPA\_PROPOGATE*  
*FIPA\_PROPOSE*  
*FIPA\_PROXY*  
*FIPA\_QUERY\_IF*  
*FIPA\_QUERY\_REF*  
*FIPA\_REFUSE*  
*FIPA\_REJECT\_PROPOSAL*  
*FIPA\_REQUEST*  
*FIPA\_REQUEST\_WHEN*  
*FIPA\_REQUEST\_WHenever*  
*FIPA\_SUBSCRIBE*  
*FIPA\_ERROR*  
*FIPA\_ZERO*  
*FIPA\_ACCEPT\_PROPOSAL*  
*FIPA\_AGREE*  
*FIPA\_CANCEL*  
*FIPA\_CALL\_FOR\_PROPOSAL*  
*FIPA\_CONFIRM*  
*FIPA\_DISCONFIRM*  
*FIPA\_FAILURE*  
*FIPA\_INFORM*  
*FIPA\_INFORM\_IF*  
*FIPA\_INFORM\_REF*  
*FIPA\_NOT\_UNDERSTOOD*  
*FIPA\_PROPOGATE*  
*FIPA\_PROPOSE*  
*FIPA\_PROXY*  
*FIPA\_QUERY\_IF*  
*FIPA\_QUERY\_REF*  
*FIPA\_REFUSE*  
*FIPA\_REJECT\_PROPOSAL*  
*FIPA\_REQUEST*  
*FIPA\_REQUEST\_WHEN*  
*FIPA\_REQUEST\_WHenever*  
*FIPA\_SUBSCRIBE*

Definition at line 41 of file fipa\_acl.h.



### 13.33.2.3 enum fipa\_protocol\_e

Enumerator:

*FIPA\_PROTOCOL\_ERROR*  
*FIPA\_PROTOCOL\_NONE*  
*FIPA\_PROTOCOL\_REQUEST*  
*FIPA\_PROTOCOL\_QUERY*  
*FIPA\_PROTOCOL\_REQUEST\_WHEN*  
*FIPA\_PROTOCOL\_CONTRACT\_NET*  
*FIPA\_PROTOCOL\_ITERATED\_CONTRACT\_NET*  
*FIPA\_PROTOCOL\_ENGLISH\_AUCTION*  
*FIPA\_PROTOCOL\_DUTCH\_AUCTION*  
*FIPA\_PROTOCOL\_BROKERING*  
*FIPA\_PROTOCOL\_RECRUITING*  
*FIPA\_PROTOCOL\_SUBSCRIBE*  
*FIPA\_PROTOCOL\_PROPOSE*  
*FIPA\_PROTOCOL\_END*

Definition at line 72 of file fipa\_acl.h.

## 13.33.3 Function Documentation

### 13.33.3.1 int fipa\_acl\_Compose (dynstring\_t \*\* msg, fipa\_acl\_message\_t \* acl)

Definition at line 1232 of file fipa\_acl.c.

References fipa\_acl\_message\_s::content, fipa\_acl\_message\_s::conversation\_id, dynstring\_Append(), dynstring\_New(), fipa\_acl\_message\_s::encoding, fipa\_agent\_identifier\_Compose(), fipa\_agent\_identifier\_set\_Compose(), fipa\_DateTime\_Compose(), fipa\_expression\_Compose(), fipa\_performative\_Compose(), fipa\_protocol\_Compose(), FIPA\_PROTOCOL\_NONE, fipa\_string\_Compose(), fipa\_acl\_message\_s::in\_reply\_to, fipa\_acl\_message\_s::language, fipa\_acl\_message\_s::ontology, fipa\_acl\_message\_s::performative, fipa\_acl\_message\_s::protocol, fipa\_acl\_message\_s::receiver, fipa\_acl\_message\_s::reply\_by, fipa\_acl\_message\_s::reply\_to, fipa\_acl\_message\_s::reply\_with, and fipa\_acl\_message\_s::sender.

Referenced by MC\_AclSend().

### 13.33.3.2 fipa\_acl\_message\_t\* fipa\_acl\_message\_Copy (fipa\_acl\_message\_t \* src)

Definition at line 86 of file fipa\_acl.c.

References fipa\_acl\_message\_s::content, fipa\_acl\_message\_s::conversation\_id, fipa\_acl\_message\_s::encoding, fipa\_acl\_message\_New(), fipa\_agent\_identifier\_Copy(), fipa\_agent\_identifier\_set\_Copy(), fipa\_DateTime\_Copy(), fipa\_expression\_Copy(), fipa\_string\_Copy(), fipa\_acl\_message\_s::in\_reply\_to, fipa\_acl\_message\_s::language, fipa\_acl\_message\_s::ontology, fipa\_acl\_message\_s::performative, fipa\_acl\_message\_s::protocol, fipa\_acl\_message\_s::receiver, fipa\_acl\_message\_s::reply\_by, fipa\_acl\_message\_s::reply\_to, fipa\_acl\_message\_s::reply\_with, and fipa\_acl\_message\_s::sender.

Referenced by MC\_AclSend().

**13.33.3.3 int fipa\_acl\_message\_Destroy (fipa\_acl\_message\_t \* *message*)**

Definition at line 65 of file fipa\_acl.c.

References fipa\_acl\_message\_s::content, fipa\_acl\_message\_s::conversation\_id, fipa\_acl\_message\_s::encoding, fipa\_agent\_identifier\_Destroy(), fipa\_agent\_identifier\_set\_Destroy(), fipa\_DateTime\_Destroy(), fipa\_expression\_Destroy(), fipa\_string\_Destroy(), fipa\_acl\_message\_s::in\_reply\_to, fipa\_acl\_message\_s::language, fipa\_acl\_message\_s::ontology, fipa\_acl\_message\_s::receiver, fipa\_acl\_message\_s::reply\_by, fipa\_acl\_message\_s::reply\_to, fipa\_acl\_message\_s::reply\_with, and fipa\_acl\_message\_s::sender.

Referenced by acc\_connection\_Thread(), and MC\_AclDestroy().

**13.33.3.4 fipa\_acl\_message\_t\* fipa\_acl\_message\_New (void)**

Definition at line 57 of file fipa\_acl.c.

Referenced by acc\_connection\_Thread(), fipa\_acl\_message\_Copy(), fipa\_Reply(), and MC\_AclNew().

**13.33.3.5 int fipa\_acl\_Parse (struct fipa\_acl\_message\_s \* *acl*, fipa\_message\_string\_p *message*)****13.33.3.6 int fipa\_agent\_identifier\_Compose (dynstring\_t \* *msg*, fipa\_agent\_identifier\_t \* *id*)**

Definition at line 1446 of file fipa\_acl.c.

References fipa\_agent\_identifier\_s::addresses, dynstring\_Append(), fipa\_agent\_identifier\_set\_Compose(), fipa\_url\_sequence\_Compose(), fipa\_agent\_identifier\_s::name, fipa\_agent\_identifier\_set\_s::num, fipa\_url\_sequence\_s::num, and fipa\_agent\_identifier\_s::resolvers.

Referenced by fipa\_acl\_Compose(), and fipa\_agent\_identifier\_set\_Compose().

**13.33.3.7 fipa\_agent\_identifier\_t\* fipa\_agent\_identifier\_Copy (fipa\_agent\_identifier\_t \* *src*)**

Definition at line 235 of file fipa\_acl.c.

References fipa\_agent\_identifier\_s::addresses, fipa\_agent\_identifier\_New(), fipa\_agent\_identifier\_set\_Copy(), fipa\_url\_sequence\_Copy(), fipa\_agent\_identifier\_s::name, and fipa\_agent\_identifier\_s::resolvers.

Referenced by fipa\_acl\_message\_Copy(), fipa\_acl\_Param\_Copy(), fipa\_agent\_identifier\_set\_Copy(), and fipa\_Reply().

**13.33.3.8 int fipa\_agent\_identifier\_Destroy (fipa\_agent\_identifier\_t \* *id*)**

Definition at line 223 of file fipa\_acl.c.

References fipa\_agent\_identifier\_s::addresses, fipa\_agent\_identifier\_set\_Destroy(), fipa\_url\_sequence\_Destroy(), fipa\_agent\_identifier\_s::name, and fipa\_agent\_identifier\_s::resolvers.

Referenced by fipa\_acl\_message\_Destroy(), fipa\_acl\_Param\_Destroy(), fipa\_agent\_identifier\_set\_Destroy(), and MC\_AclSetSender().

**13.33.3.9 fipa\_agent\_identifier\_t\* fipa\_agent\_identifier\_New (void)**

Definition at line 215 of file fipa\_acl.c.

Referenced by `fipa_agent_identifier_Copy()`, `fipa_envelope_ParseAgentIdentifier()`, `MC_AclAddReceiver()`, `MC_AclAddReplyTo()`, and `MC_AclSetSender()`.

### **13.33.3.10 `int fipa_agent_identifier_Parse (fipa_agent_identifier_p * aid, fipa_message_string_p message)`**

Definition at line 1032 of file `fipa_acl.c`.

References `CHECK_NULL`, `fipa_word_s::content`, `fipa_agent_identifier_set_Parse()`, `fipa_GetAtom()`, `fipa_url_sequence_Parse()`, `fipa_word_Destroy()`, `fipa_word_Parse()`, `MC_ERR_PARSE`, `MC_SUCCESS`, and `fipa_message_string_s::parse`.

Referenced by `fipa_agent_identifier_set_Parse()`, and `fipa_message_parameter_Parse()`.

### **13.33.3.11 `int fipa_agent_identifier_set_Compose (dynstring_t * msg, fipa_agent_identifier_set_t * ids)`**

Definition at line 1433 of file `fipa_acl.c`.

References `dynstring_Append()`, `fipa_agent_identifier_Compose()`, `fipa_agent_identifier_set_s::fipa_agent_identifiers`, and `fipa_agent_identifier_set_s::num`.

Referenced by `fipa_acl_Compose()`, and `fipa_agent_identifier_Compose()`.

### **13.33.3.12 `fipa_agent_identifier_set_t* fipa_agent_identifier_set_Copy (fipa_agent_identifier_set_t * src)`**

Definition at line 194 of file `fipa_acl.c`.

References `fipa_agent_identifier_Copy()`, `fipa_agent_identifier_set_New()`, `fipa_agent_identifier_set_s::fipa_agent_identifiers`, `fipa_agent_identifier_set_s::num`, and `fipa_agent_identifier_set_s::retain_order`.

Referenced by `fipa_acl_message_Copy()`, `fipa_acl_Param_Copy()`, `fipa_agent_identifier_Copy()`, and `fipa_Reply()`.

### **13.33.3.13 `int fipa_agent_identifier_set_Destroy (fipa_agent_identifier_set_t * idset)`**

Definition at line 182 of file `fipa_acl.c`.

References `fipa_agent_identifier_Destroy()`, `fipa_agent_identifier_set_s::fipa_agent_identifiers`, and `fipa_agent_identifier_set_s::num`.

Referenced by `fipa_acl_message_Destroy()`, `fipa_acl_Param_Destroy()`, and `fipa_agent_identifier_Destroy()`.

### **13.33.3.14 `fipa_agent_identifier_set_t* fipa_agent_identifier_set_New (void)`**

Definition at line 174 of file `fipa_acl.c`.

Referenced by `fipa_agent_identifier_set_Copy()`, `fipa_envelope_HandleIntendedReceiver()`, `fipa_envelope_HandleTo()`, `fipa_Reply()`, `MC_AclAddReceiver()`, and `MC_AclAddReplyTo()`.

### 13.33.3.15 **int fipa\_agent\_identifier\_set\_Parse (fipa\_agent\_identifier\_set\_p \* *agent\_ids*, fipa\_message\_string\_p *message*)**

Definition at line 1191 of file fipa\_acl.c.

References fipa\_word\_s::content, fipa\_agent\_identifier\_Parse(), fipa\_GetAtom(), fipa\_word\_Parse(), MC\_ERR\_PARSE, MC\_SUCCESS, and fipa\_agent\_identifier\_set\_s::retain\_order.

Referenced by fipa\_agent\_identifier\_Parse(), and fipa\_message\_parameter\_Parse().

### 13.33.3.16 **int fipa\_CheckNextToken (const fipa\_message\_string\_p *message*, const char \* *token*)**

Definition at line 694 of file fipa\_acl.c.

References fipa\_message\_string\_s::parse.

Referenced by fipa\_expression\_Parse().

### 13.33.3.17 **int fipa\_DateTime\_Compose (dynstring\_t \* *msg*, fipa\_DateTime\_t \* *date*)**

Definition at line 1519 of file fipa\_acl.c.

References buf, fipa\_DateTime\_s::day, dynstring\_Append(), fipa\_DateTime\_s::hour, fipa\_DateTime\_s::millisecond, fipa\_DateTime\_s::minute, fipa\_DateTime\_s::month, fipa\_DateTime\_s::second, fipa\_DateTime\_s::sign, and fipa\_DateTime\_s::year.

Referenced by fipa\_acl\_Compose(), and fipa\_expression\_Compose().

### 13.33.3.18 **fipa\_DateTime\_t\* fipa\_DateTime\_Copy (fipa\_DateTime\_t \* *src*)**

Definition at line 398 of file fipa\_acl.c.

References fipa\_DateTime\_New().

Referenced by fipa\_acl\_envelope\_Received\_Copy(), fipa\_acl\_message\_Copy(), fipa\_acl\_Param\_Copy(), and fipa\_expression\_Copy().

### 13.33.3.19 **int fipa\_DateTime\_Destroy (fipa\_DateTime\_t \* *dt*)**

Definition at line 391 of file fipa\_acl.c.

Referenced by fipa\_acl\_envelope\_Received\_Destroy(), fipa\_acl\_message\_Destroy(), fipa\_acl\_Param\_Destroy(), and fipa\_expression\_Destroy().

### 13.33.3.20 **fipa\_DateTime\_t\* fipa\_DateTime\_New (void)**

Definition at line 383 of file fipa\_acl.c.

Referenced by fipa\_DateTime\_Copy().

### 13.33.3.21 **int fipa\_datetime\_Parse (fipa\_DateTime\_p \* *datetime*, fipa\_message\_string\_p *message*)**

Definition at line 849 of file fipa\_acl.c.

References `buf`, `fipa_GetWholeToken()`, `MC_ERR_PARSE`, `MC_SUCCESS`, `fipa_message_string_s::parse`, and `fipa_DateTime_s::sign`.

Referenced by `fipa_envelope_HandleDate()`, `fipa_envelope_HandleReceived()`, `fipa_expression_Parse()`, and `fipa_message_parameter_Parse()`.

### 13.33.3.22 `int fipa_expression_Compose (dynstring_t * msg, fipa_expression_t * expr)`

Definition at line 1472 of file `fipa_acl.c`.

References `fipa_expression_s::content`, `fipa_expression_s::content_u::datetime`, `fipa_expression_s::content_u::expression`, `fipa_DateTime_Compose()`, `FIPA_EXPR_DATETIME`, `FIPA_EXPR_EXPRESSION`, `FIPA_EXPR_NUMBER`, `FIPA_EXPR_STRING`, `FIPA_EXPR_WORD`, `fipa_expression_Compose()`, `fipa_number_Compose()`, `fipa_string_Compose()`, `fipa_word_Compose()`, `MC_ERR_PARSE`, `fipa_expression_s::content_u::number`, `fipa_expression_s::content_u::string`, `fipa_expression_s::type`, and `fipa_expression_s::content_u::word`.

Referenced by `fipa_acl_Compose()`, and `fipa_expression_Compose()`.

### 13.33.3.23 `fipa_expression_t* fipa_expression_Copy (fipa_expression_t * src)`

Definition at line 286 of file `fipa_acl.c`.

References `fipa_expression_s::content`, `fipa_expression_s::content_u::datetime`, `fipa_expression_s::content_u::expression`, `fipa_DateTime_Copy()`, `FIPA_EXPR_DATETIME`, `FIPA_EXPR_EXPRESSION`, `FIPA_EXPR_NUMBER`, `FIPA_EXPR_STRING`, `FIPA_EXPR_WORD`, `fipa_expression_Copy()`, `fipa_expression_Destroy()`, `fipa_expression_New()`, `fipa_number_Copy()`, `fipa_string_Copy()`, `fipa_word_Copy()`, `fipa_expression_s::content_u::number`, `fipa_expression_s::content_u::string`, `fipa_expression_s::type`, and `fipa_expression_s::content_u::word`.

Referenced by `fipa_acl_message_Copy()`, `fipa_expression_Copy()`, and `fipa_Reply()`.

### 13.33.3.24 `int fipa_expression_Destroy (fipa_expression_t * expr)`

Definition at line 255 of file `fipa_acl.c`.

References `fipa_expression_s::content`, `fipa_expression_s::content_u::datetime`, `fipa_expression_s::content_u::expression`, `fipa_DateTime_Destroy()`, `FIPA_EXPR_DATETIME`, `FIPA_EXPR_EXPRESSION`, `FIPA_EXPR_NUMBER`, `FIPA_EXPR_STRING`, `FIPA_EXPR_WORD`, `fipa_expression_Destroy()`, `fipa_number_Destroy()`, `fipa_string_Destroy()`, `fipa_word_Destroy()`, `FREEMEM`, `fipa_expression_s::content_u::number`, `fipa_expression_s::content_u::string`, `fipa_expression_s::type`, and `fipa_expression_s::content_u::word`.

Referenced by `fipa_acl_message_Destroy()`, `fipa_expression_Copy()`, and `fipa_expression_Destroy()`.

### 13.33.3.25 `fipa_expression_t* fipa_expression_New (void)`

Definition at line 247 of file `fipa_acl.c`.

Referenced by `fipa_expression_Copy()`, and `MC_AclSetConversationID()`.

### 13.33.3.26 `int fipa_expression_Parse (fipa_expression_t ** expression, fipa_message_string_p message)`

Definition at line 713 of file `fipa_acl.c`.

References `fipa_CheckNextToken()`, `fipa_datetime_Parse()`, `FIPA_EXPR_DATETIME`, `FIPA_EXPR_EXPRESSION`, `FIPA_EXPR_STRING`, `FIPA_EXPR_WORD`, `fipa_expression_Parse()`, `fipa_GetAtom()`, `fipa_string_Parse()`, `fipa_word_Parse()`, `MC_ERR_PARSE`, `MC_SUCCESS`, and `fipa_expression_s::type`.

Referenced by `fipa_expression_Parse()`, and `fipa_message_parameter_Parse()`.

### 13.33.3.27 `int fipa_GetAtom (fipa_message_string_p message, char expected_atom)`

Definition at line 629 of file `fipa_acl.c`.

References `MC_ERR_PARSE`, `MC_SUCCESS`, and `fipa_message_string_s::parse`.

Referenced by `fipa_acl_Parse()`, `fipa_agent_identifier_Parse()`, `fipa_agent_identifier_set_Parse()`, `fipa_expression_Parse()`, `fipa_message_parameter_Parse()`, `fipa_string_Parse()`, and `fipa_url_sequence_Parse()`.

### 13.33.3.28 `int fipa_GetNextWord (char ** word, const fipa_message_string_p message)`

Definition at line 764 of file `fipa_acl.c`.

References `ERR`, `MC_SUCCESS`, and `fipa_message_string_s::parse`.

### 13.33.3.29 `int fipa_GetWholeToken (char ** word, const fipa_message_string_p message)`

Definition at line 812 of file `fipa_acl.c`.

References `MC_SUCCESS`, and `fipa_message_string_s::parse`.

Referenced by `fipa_datetime_Parse()`.

### 13.33.3.30 `int fipa_message_parameter_Parse (struct fipa_acl_message_s * acl, fipa_message_string_p message)`

### 13.33.3.31 `fipa_message_string_t* fipa_message_string_Copy (fipa_message_string_t * src)`

Definition at line 128 of file `fipa_acl.c`.

References `fipa_message_string_s::message`, and `fipa_message_string_s::parse`.

### 13.33.3.32 `int fipa_message_string_Destroy (fipa_message_string_t * message)`

Definition at line 118 of file `fipa_acl.c`.

References `fipa_message_string_s::message`.

Referenced by `acc_connection_Thread()`.

### 13.33.3.33 `fipa_message_string_t* fipa_message_string_New (void)`

Definition at line 110 of file `fipa_acl.c`.

Referenced by `acc_connection_Thread()`.

### 13.33.3.34 `int fipa_message_type_Parse (enum fipa_performative_e * performative, fipa_message_string_p message)`

Definition at line 567 of file `fipa_acl.c`.

References `fipa_word_s::content`, `FIPA_ACCEPT_PROPOSAL`, `FIPA_AGREE`, `FIPA_CALL_FOR_PROPOSAL`, `FIPA_CANCEL`, `FIPA_CONFIRM`, `FIPA_DISCONFIRM`, `FIPA_FAILURE`, `FIPA_INFORM`, `FIPA_INFORM_IF`, `FIPA_INFORM_REF`, `FIPA_NOT_UNDERSTOOD`, `FIPA_PROPOGATE`, `FIPA_PROPOSE`, `FIPA_PROXY`, `FIPA_QUERY_IF`, `FIPA_QUERY_REF`, `FIPA_REFUSE`, `FIPA_REJECT_PROPOSAL`, `FIPA_REQUEST`, `FIPA_REQUEST_WHEN`, `FIPA_REQUEST_WHENEVER`, `FIPA_SUBSCRIBE`, `fipa_word_Destroy()`, `fipa_word_Parse()`, and `MC_ERR_PARSE`.

Referenced by `fipa_acl_Parse()`.

### 13.33.3.35 `int fipa_number_Compose (dynstring_t * msg, fipa_number_t * number)`

Definition at line 1547 of file `fipa_acl.c`.

References `dynstring_Append()`, and `fipa_number_s::str`.

Referenced by `fipa_expression_Compose()`.

### 13.33.3.36 `fipa_number_t* fipa_number_Copy (fipa_number_t * src)`

Definition at line 454 of file `fipa_acl.c`.

References `fipa_number_New()`, and `fipa_number_s::str`.

Referenced by `fipa_expression_Copy()`.

### 13.33.3.37 `int fipa_number_Destroy (fipa_number_t * number)`

Definition at line 444 of file `fipa_acl.c`.

References `fipa_number_s::str`.

Referenced by `fipa_expression_Destroy()`.

### 13.33.3.38 `fipa_number_t* fipa_number_New (void)`

Definition at line 436 of file `fipa_acl.c`.

Referenced by `fipa_number_Copy()`.

### 13.33.3.39 `int fipa_performative_Compose (dynstring_t * msg, enum fipa_performative_e performative)`

Definition at line 1345 of file `fipa_acl.c`.

References `dynstring_Append()`, `FIPA_ACCEPT_PROPOSAL`, `FIPA_AGREE`, `FIPA_CALL_FOR_PROPOSAL`, `FIPA_CANCEL`, `FIPA_CONFIRM`, `FIPA_DISCONFIRM`, `FIPA_FAILURE`,

FIPA\_INFORM, FIPA\_INFORM\_IF, FIPA\_INFORM\_REF, FIPA\_NOT\_UNDERSTOOD, FIPA\_PROPOGATE, FIPA\_PROPOSE, FIPA\_PROXY, FIPA\_QUERY\_IF, FIPA\_QUERY\_REF, FIPA\_REFUSE, FIPA\_REJECT\_PROPOSAL, FIPA\_REQUEST, FIPA\_REQUEST\_WHEN, FIPA\_REQUEST\_WHENEVER, FIPA\_SUBSCRIBE, and MC\_ERR\_PARSE.

Referenced by fipa\_acl\_Compose().

#### 13.33.3.40 **int fipa\_protocol\_Compose** (dynstring\_t \* *msg*, enum fipa\_protocol\_e *protocol*)

Definition at line 1302 of file fipa\_acl.c.

References dynstring\_Append(), FIPA\_PROTOCOL\_BROKERING, FIPA\_PROTOCOL\_CONTRACT\_NET, FIPA\_PROTOCOL\_DUTCH\_AUCTION, FIPA\_PROTOCOL\_ENGLISH\_AUCTION, FIPA\_PROTOCOL\_ITERATED\_CONTRACT\_NET, FIPA\_PROTOCOL\_PROPOSE, FIPA\_PROTOCOL\_QUERY, FIPA\_PROTOCOL\_RECRUITING, FIPA\_PROTOCOL\_REQUEST, FIPA\_PROTOCOL\_REQUEST\_WHEN, FIPA\_PROTOCOL\_SUBSCRIBE, and MC\_ERR\_PARSE.

Referenced by fipa\_acl\_Compose().

#### 13.33.3.41 **int fipa\_protocol\_type\_Parse** (enum fipa\_protocol\_e \* *protocol*, fipa\_message\_string\_p *message*)

Definition at line 527 of file fipa\_acl.c.

References fipa\_word\_s::content, FIPA\_PROTOCOL\_BROKERING, FIPA\_PROTOCOL\_CONTRACT\_NET, FIPA\_PROTOCOL\_DUTCH\_AUCTION, FIPA\_PROTOCOL\_ENGLISH\_AUCTION, FIPA\_PROTOCOL\_ITERATED\_CONTRACT\_NET, FIPA\_PROTOCOL\_PROPOSE, FIPA\_PROTOCOL\_QUERY, FIPA\_PROTOCOL\_RECRUITING, FIPA\_PROTOCOL\_REQUEST, FIPA\_PROTOCOL\_REQUEST\_WHEN, FIPA\_PROTOCOL\_SUBSCRIBE, fipa\_word\_Destroy(), fipa\_word\_Parse(), and MC\_ERR\_PARSE.

Referenced by fipa\_message\_parameter\_Parse().

#### 13.33.3.42 **struct fipa\_acl\_message\_s\* fipa\_Reply** (struct fipa\_acl\_message\_s \* *acl*) **[read]**

Definition at line 1555 of file fipa\_acl.c.

References fipa\_acl\_message\_s::conversation\_id, fipa\_acl\_message\_New(), fipa\_agent\_identifier\_Copy(), fipa\_agent\_identifier\_set\_Copy(), fipa\_agent\_identifier\_set\_New(), fipa\_agent\_identifier\_set\_s::fipa\_agent\_identifiers, fipa\_expression\_Copy(), fipa\_agent\_identifier\_set\_s::num, fipa\_acl\_message\_s::protocol, fipa\_acl\_message\_s::receiver, fipa\_acl\_message\_s::reply\_to, fipa\_agent\_identifier\_set\_s::retain\_order, and fipa\_acl\_message\_s::sender.

Referenced by MC\_AclReply().

#### 13.33.3.43 **int fipa\_string\_Compose** (dynstring\_t \* *msg*, fipa\_string\_t \* *string*)

Definition at line 1510 of file fipa\_acl.c.

References fipa\_string\_s::content, and dynstring\_Append().

Referenced by fipa\_acl\_Compose(), and fipa\_expression\_Compose().



#### 13.33.3.44 **fipa\_string\_t\* fipa\_string\_Copy (fipa\_string\_t \* *src*)**

Definition at line 373 of file fipa\_acl.c.

References fipa\_string\_s::content, and fipa\_string\_New().

Referenced by fipa\_acl\_message\_Copy(), and fipa\_expression\_Copy().

#### 13.33.3.45 **int fipa\_string\_Destroy (fipa\_string\_t \* *str*)**

Definition at line 363 of file fipa\_acl.c.

References fipa\_string\_s::content.

Referenced by fipa\_acl\_message\_Destroy(), fipa\_expression\_Destroy(), and MC\_AclSetContent().

#### 13.33.3.46 **fipa\_string\_t\* fipa\_string\_New (void)**

Definition at line 355 of file fipa\_acl.c.

Referenced by fipa\_string\_Copy(), MC\_AclSetContent(), and MC\_AclSetConversationID().

#### 13.33.3.47 **int fipa\_string\_Parse (fipa\_string\_p \* *fipa\_string*, fipa\_message\_string\_p *message*)**

Definition at line 987 of file fipa\_acl.c.

References fipa\_string\_s::content, fipa\_GetAtom(), MC\_ERR\_PARSE, MC\_SUCCESS, and fipa\_message\_string\_s::parse.

Referenced by fipa\_expression\_Parse(), and fipa\_message\_parameter\_Parse().

#### 13.33.3.48 **int fipa\_url\_Compose (dynstring\_t \* *msg*, fipa\_url\_t \* *url*)**

Definition at line 1539 of file fipa\_acl.c.

References dynstring\_Append(), and fipa\_url\_s::str.

Referenced by fipa\_url\_sequence\_Compose().

#### 13.33.3.49 **fipa\_url\_t\* fipa\_url\_Copy (fipa\_url\_t \* *src*)**

Definition at line 426 of file fipa\_acl.c.

References fipa\_url\_New(), and fipa\_url\_s::str.

Referenced by fipa\_acl\_envelope\_Received\_Copy(), and fipa\_url\_sequence\_Copy().

#### 13.33.3.50 **int fipa\_url\_Destroy (fipa\_url\_t \* *url*)**

Definition at line 416 of file fipa\_acl.c.

References fipa\_url\_s::str.

Referenced by fipa\_acl\_envelope\_Received\_Destroy(), and fipa\_url\_sequence\_Destroy().

**13.33.3.51 fipa\_url\_t\* fipa\_url\_New (void)**

Definition at line 408 of file fipa\_acl.c.

Referenced by fipa\_envelope\_HandleReceived(), fipa\_envelope\_ParseAddresses(), fipa\_url\_Copy(), MC\_AclAddReceiver(), MC\_AclAddReplyTo(), and MC\_AclSetSender().

**13.33.3.52 int fipa\_url\_Parse (fipa\_url\_p \* url, fipa\_message\_string\_p message)**

Definition at line 1171 of file fipa\_acl.c.

References fipa\_word\_s::content, fipa\_word\_Destroy(), and fipa\_word\_Parse().

Referenced by fipa\_url\_sequence\_Parse().

**13.33.3.53 int fipa\_url\_sequence\_Compose (dynstring\_t \* msg, fipa\_url\_sequence\_t \* urls)**

Definition at line 1420 of file fipa\_acl.c.

References dynstring\_Append(), fipa\_url\_Compose(), fipa\_url\_sequence\_s::num, and fipa\_url\_sequence\_s::urls.

Referenced by fipa\_agent\_identifier\_Compose().

**13.33.3.54 fipa\_url\_sequence\_t\* fipa\_url\_sequence\_Copy (fipa\_url\_sequence\_t \* src)**

Definition at line 158 of file fipa\_acl.c.

References fipa\_url\_Copy(), fipa\_url\_sequence\_New(), fipa\_url\_sequence\_s::num, and fipa\_url\_sequence\_s::urls.

Referenced by fipa\_agent\_identifier\_Copy().

**13.33.3.55 int fipa\_url\_sequence\_Destroy (fipa\_url\_sequence\_t \* sequence)**

Definition at line 146 of file fipa\_acl.c.

References fipa\_url\_Destroy(), fipa\_url\_sequence\_s::num, and fipa\_url\_sequence\_s::urls.

Referenced by fipa\_agent\_identifier\_Destroy().

**13.33.3.56 fipa\_url\_sequence\_t\* fipa\_url\_sequence\_New (void)**

Definition at line 138 of file fipa\_acl.c.

Referenced by fipa\_url\_sequence\_Copy(), fipa\_url\_sequence\_Parse(), MC\_AclAddReceiver(), MC\_AclAddReplyTo(), and MC\_AclSetSender().

**13.33.3.57 int fipa\_url\_sequence\_Parse (fipa\_url\_sequence\_p \* urls, fipa\_message\_string\_p message)**

Definition at line 1139 of file fipa\_acl.c.

References fipa\_word\_s::content, fipa\_GetAtom(), fipa\_url\_Parse(), fipa\_url\_sequence\_New(), fipa\_word\_Destroy(), fipa\_word\_Parse(), and MC\_ERR\_PARSE.

Referenced by `fipa_agent_identifier_Parse()`.

#### **13.33.3.58 `int fipa_word_Compose (dynstring_t * msg, fipa_word_t * word)`**

Definition at line 1502 of file `fipa_acl.c`.

References `fipa_word_s::content`, and `dynstring_Append()`.

Referenced by `fipa_expression_Compose()`.

#### **13.33.3.59 `fipa_word_t* fipa_word_Copy (fipa_word_t * src)`**

Definition at line 345 of file `fipa_acl.c`.

References `fipa_word_s::content`, and `fipa_word_New()`.

Referenced by `fipa_expression_Copy()`.

#### **13.33.3.60 `int fipa_word_Destroy (fipa_word_t * word)`**

Definition at line 335 of file `fipa_acl.c`.

References `fipa_word_s::content`.

Referenced by `fipa_agent_identifier_Parse()`, `fipa_expression_Destroy()`, `fipa_message_parameter_Parse()`, `fipa_message_type_Parse()`, `fipa_protocol_type_Parse()`, `fipa_url_Parse()`, and `fipa_url_sequence_Parse()`.

#### **13.33.3.61 `fipa_word_t* fipa_word_New (void)`**

Definition at line 327 of file `fipa_acl.c`.

Referenced by `fipa_word_Copy()`.

#### **13.33.3.62 `int fipa_word_Parse (fipa_word_t ** word, fipa_message_string_p message)`**

Definition at line 652 of file `fipa_acl.c`.

References `CHECK_NULL`, `MC_ERR_PARSE`, `MC_SUCCESS`, and `fipa_message_string_s::parse`.

Referenced by `fipa_agent_identifier_Parse()`, `fipa_agent_identifier_set_Parse()`, `fipa_expression_Parse()`, `fipa_message_parameter_Parse()`, `fipa_message_type_Parse()`, `fipa_protocol_type_Parse()`, `fipa_url_Parse()`, and `fipa_url_sequence_Parse()`.

## 13.34 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/fipa\_acl\_envelope.h File Reference

```
#include "fipa_acl.h"
```

### Data Structures

- struct [fipa\\_acl\\_envelope\\_Received\\_s](#)
- struct [fipa\\_acl\\_Param\\_s](#)
- struct [fipa\\_acl\\_envelope\\_s](#)

### Typedefs

- typedef struct [fipa\\_acl\\_envelope\\_Received\\_s](#) [fipa\\_acl\\_envelope\\_Received\\_t](#)
- typedef struct [fipa\\_acl\\_Param\\_s](#) [fipa\\_acl\\_Param\\_t](#)
- typedef struct [fipa\\_acl\\_envelope\\_s](#) [fipa\\_acl\\_envelope\\_t](#)
- typedef [fipa\\_acl\\_envelope\\_t](#) \* [fipa\\_acl\\_envelope\\_p](#)

### Functions

- [fipa\\_acl\\_envelope\\_Received\\_t](#) \* [fipa\\_acl\\_envelope\\_Received\\_New](#) (void)
- int [fipa\\_acl\\_envelope\\_Received\\_Destroy](#) ([fipa\\_acl\\_envelope\\_Received\\_t](#) \*received)
- [fipa\\_acl\\_envelope\\_Received\\_t](#) \* [fipa\\_acl\\_envelope\\_Received\\_Copy](#) ([fipa\\_acl\\_envelope\\_Received\\_t](#) \*received)
- [fipa\\_acl\\_Param\\_t](#) \* [fipa\\_acl\\_Param\\_New](#) (void)
- int [fipa\\_acl\\_Param\\_Destroy](#) ([fipa\\_acl\\_Param\\_t](#) \*param)
- [fipa\\_acl\\_Param\\_t](#) \* [fipa\\_acl\\_Param\\_Copy](#) ([fipa\\_acl\\_Param\\_t](#) \*param)
- [fipa\\_acl\\_envelope\\_t](#) \* [fipa\\_acl\\_envelope\\_New](#) (void)
- int [fipa\\_acl\\_envelope\\_Destroy](#) ([fipa\\_acl\\_envelope\\_t](#) \*envelope)
- [fipa\\_acl\\_envelope\\_t](#) \* [fipa\\_acl\\_envelope\\_Copy](#) ([fipa\\_acl\\_envelope\\_t](#) \*envelope)
- int [fipa\\_envelope\\_Parse](#) (struct [fipa\\_acl\\_envelope\\_s](#) \*envelope, const char \*message)
- int [fipa\\_envelope\\_HandleEnvelope](#) (struct [fipa\\_acl\\_envelope\\_s](#) \*envelope, [mxml\\_node\\_t](#) \*node)
- int [fipa\\_envelope\\_HandleParams](#) (struct [fipa\\_acl\\_envelope\\_s](#) \*envelope, [mxml\\_node\\_t](#) \*node)
- int [fipa\\_envelope\\_HandleTo](#) (struct [fipa\\_acl\\_envelope\\_s](#) \*envelope, [mxml\\_node\\_t](#) \*param\_node, int cur\_param)
- int [fipa\\_envelope\\_HandleFrom](#) (struct [fipa\\_acl\\_envelope\\_s](#) \*envelope, [mxml\\_node\\_t](#) \*param\_node, int cur\_param)
- int [fipa\\_envelope\\_HandleComments](#) (struct [fipa\\_acl\\_envelope\\_s](#) \*envelope, [mxml\\_node\\_t](#) \*param\_node, int cur\_param)
- int [fipa\\_envelope\\_HandleAclRepresentation](#) (struct [fipa\\_acl\\_envelope\\_s](#) \*envelope, [mxml\\_node\\_t](#) \*param\_node, int cur\_param)
- int [fipa\\_envelope\\_HandlePayloadLength](#) (struct [fipa\\_acl\\_envelope\\_s](#) \*envelope, [mxml\\_node\\_t](#) \*param\_node, int cur\_param)
- int [fipa\\_envelope\\_HandlePayloadEncoding](#) (struct [fipa\\_acl\\_envelope\\_s](#) \*envelope, [mxml\\_node\\_t](#) \*param\_node, int cur\_param)
- int [fipa\\_envelope\\_HandleDate](#) (struct [fipa\\_acl\\_envelope\\_s](#) \*envelope, [mxml\\_node\\_t](#) \*param\_node, int cur\_param)
- int [fipa\\_envelope\\_HandleIntendedReceiver](#) (struct [fipa\\_acl\\_envelope\\_s](#) \*envelope, [mxml\\_node\\_t](#) \*param\_node, int cur\_param)

- [int fipa\\_envelope\\_HandleReceived](#) (struct [fipa\\_acl\\_envelope\\_s](#) \*envelope, [mxml\\_node\\_t](#) \*param\_node, int cur\_param)
- [int fipa\\_envelope\\_ParseAgentIdentifier](#) (struct [fipa\\_agent\\_identifier\\_s](#) \*\*aid, [mxml\\_node\\_t](#) \*agent\_identifier\_node)
- [int fipa\\_envelope\\_ParseAddresses](#) (struct [fipa\\_agent\\_identifier\\_s](#) \*aid, [mxml\\_node\\_t](#) \*addresses\_node)
- [int fipa\\_envelope\\_ParseResolvers](#) (struct [fipa\\_agent\\_identifier\\_s](#) \*aid, [mxml\\_node\\_t](#) \*resolvers\_node)
- [char \\* fipa\\_envelope\\_Compose](#) ([fipa\\_acl\\_message\\_t](#) \*fipa\_acl)
- [mxml\\_node\\_t \\* fipa\\_envelope\\_Compose\\_\\_envelope](#) ([fipa\\_acl\\_message\\_t](#) \*fipa\_acl)
- [mxml\\_node\\_t \\* fipa\\_envelope\\_Compose\\_\\_params](#) ([fipa\\_acl\\_message\\_t](#) \*fipa\_acl)
- [mxml\\_node\\_t \\* fipa\\_envelope\\_Compose\\_\\_to](#) ([fipa\\_acl\\_message\\_t](#) \*fipa\_acl)
- [mxml\\_node\\_t \\* fipa\\_envelope\\_Compose\\_\\_from](#) ([fipa\\_acl\\_message\\_t](#) \*fipa\_acl)
- [mxml\\_node\\_t \\* fipa\\_envelope\\_Compose\\_\\_acl\\_representation](#) ([fipa\\_acl\\_message\\_t](#) \*fipa\_acl)
- [mxml\\_node\\_t \\* fipa\\_envelope\\_Compose\\_\\_payload\\_encoding](#) ([fipa\\_acl\\_message\\_t](#) \*fipa\_acl)
- [mxml\\_node\\_t \\* fipa\\_envelope\\_Compose\\_\\_date](#) ([fipa\\_acl\\_message\\_t](#) \*fipa\_acl)
- [mxml\\_node\\_t \\* fipa\\_envelope\\_Compose\\_\\_intended\\_receiver](#) ([fipa\\_acl\\_message\\_t](#) \*fipa\_acl)

### 13.34.1 Typedef Documentation

#### 13.34.1.1 typedef [fipa\\_acl\\_envelope\\_t](#)\* [fipa\\_acl\\_envelope\\_p](#)

Definition at line 45 of file [fipa\\_acl\\_envelope.h](#).

#### 13.34.1.2 typedef struct [fipa\\_acl\\_envelope\\_Received\\_s](#) [fipa\\_acl\\_envelope\\_Received\\_t](#)

#### 13.34.1.3 typedef struct [fipa\\_acl\\_envelope\\_s](#) [fipa\\_acl\\_envelope\\_t](#)

#### 13.34.1.4 typedef struct [fipa\\_acl\\_Param\\_s](#) [fipa\\_acl\\_Param\\_t](#)

### 13.34.2 Function Documentation

#### 13.34.2.1 [fipa\\_acl\\_envelope\\_t](#)\* [fipa\\_acl\\_envelope\\_Copy](#) ([fipa\\_acl\\_envelope\\_t](#) \* *envelope*)

Definition at line 150 of file [fipa\\_envelope.c](#).

References [fipa\\_acl\\_envelope\\_New\(\)](#), [fipa\\_acl\\_Param\\_Copy\(\)](#), [fipa\\_acl\\_envelope\\_s::num\\_params](#), and [fipa\\_acl\\_envelope\\_s::params](#).

#### 13.34.2.2 int [fipa\\_acl\\_envelope\\_Destroy](#) ([fipa\\_acl\\_envelope\\_t](#) \* *envelope*)

Definition at line 137 of file [fipa\\_envelope.c](#).

References [fipa\\_acl\\_Param\\_Destroy\(\)](#), [fipa\\_acl\\_envelope\\_s::num\\_params](#), and [fipa\\_acl\\_envelope\\_s::params](#).

Referenced by [acc\\_connection\\_Thread\(\)](#).

**13.34.2.3 fipa\_acl\_envelope\_t\* fipa\_acl\_envelope\_New (void)**

Definition at line 129 of file fipa\_envelope.c.

Referenced by acc\_connection\_Thread(), and fipa\_acl\_envelope\_Copy().

**13.34.2.4 fipa\_acl\_envelope\_Received\_t\* fipa\_acl\_envelope\_Received\_Copy (fipa\_acl\_envelope\_Received\_t \* *received*)**

Definition at line 69 of file fipa\_envelope.c.

References fipa\_acl\_envelope\_Received\_New(), fipa\_DateTime\_Copy(), fipa\_url\_Copy(), fipa\_acl\_envelope\_Received\_s::received\_by, fipa\_acl\_envelope\_Received\_s::received\_date, fipa\_acl\_envelope\_Received\_s::received\_from, fipa\_acl\_envelope\_Received\_s::received\_id, and fipa\_acl\_envelope\_Received\_s::received\_via.

Referenced by fipa\_acl\_Param\_Copy().

**13.34.2.5 int fipa\_acl\_envelope\_Received\_Destroy (fipa\_acl\_envelope\_Received\_t \* *received*)**

Definition at line 57 of file fipa\_envelope.c.

References fipa\_DateTime\_Destroy(), fipa\_url\_Destroy(), fipa\_acl\_envelope\_Received\_s::received\_by, fipa\_acl\_envelope\_Received\_s::received\_date, fipa\_acl\_envelope\_Received\_s::received\_from, fipa\_acl\_envelope\_Received\_s::received\_id, and fipa\_acl\_envelope\_Received\_s::received\_via.

Referenced by fipa\_acl\_Param\_Destroy().

**13.34.2.6 fipa\_acl\_envelope\_Received\_t\* fipa\_acl\_envelope\_Received\_New (void)**

Definition at line 48 of file fipa\_envelope.c.

Referenced by fipa\_acl\_envelope\_Received\_Copy(), and fipa\_envelope\_HandleReceived().

**13.34.2.7 fipa\_acl\_Param\_t\* fipa\_acl\_Param\_Copy (fipa\_acl\_Param\_t \* *param*)**

Definition at line 110 of file fipa\_envelope.c.

References fipa\_acl\_Param\_s::acl\_representation, fipa\_acl\_Param\_s::comments, fipa\_acl\_Param\_s::date, fipa\_acl\_envelope\_Received\_Copy(), fipa\_acl\_Param\_New(), fipa\_agent\_identifier\_Copy(), fipa\_agent\_identifier\_set\_Copy(), fipa\_DateTime\_Copy(), fipa\_acl\_Param\_s::from, fipa\_acl\_Param\_s::intended\_receiver, fipa\_acl\_Param\_s::payload\_encoding, fipa\_acl\_Param\_s::payload\_length, fipa\_acl\_Param\_s::received, and fipa\_acl\_Param\_s::to.

Referenced by fipa\_acl\_envelope\_Copy().

**13.34.2.8 int fipa\_acl\_Param\_Destroy (fipa\_acl\_Param\_t \* *param*)**

Definition at line 93 of file fipa\_envelope.c.

References fipa\_acl\_Param\_s::acl\_representation, fipa\_acl\_Param\_s::comments, fipa\_acl\_Param\_s::date, fipa\_acl\_envelope\_Received\_Destroy(), fipa\_agent\_identifier\_Destroy(), fipa\_agent\_identifier\_set\_Destroy(), fipa\_DateTime\_Destroy(), fipa\_acl\_Param\_s::from, fipa\_acl\_Param\_s::intended\_receiver, fipa\_acl\_Param\_s::payload\_encoding, fipa\_acl\_Param\_s::payload\_length, fipa\_acl\_Param\_s::received, and fipa\_acl\_Param\_s::to.

Referenced by `fipa_acl_envelope_Destroy()`.

#### **13.34.2.9 `fipa_acl_Param_t* fipa_acl_Param_New (void)`**

Definition at line 85 of file `fipa_envelope.c`.

Referenced by `fipa_acl_Param_Copy()`, and `fipa_envelope_HandleTo()`.

#### **13.34.2.10 `char* fipa_envelope_Compose (fipa_acl_message_t * fipa_acl)`**

Definition at line 869 of file `fipa_envelope.c`.

References `fipa_envelope_Compose__envelope()`, `MXML_ADD_AFTER`, `MXML_ADD_TO_PARENT`, `MXML_NO_CALLBACK`, `mxmlAdd()`, `mxmlDelete()`, `mxmlLoadString()`, `mxmlSaveAllocString()`, and `node`.

Referenced by `MC_AclSend()`.

#### **13.34.2.11 `mxml_node_t* fipa_envelope_Compose__acl_representation (fipa_acl_message_t * fipa_acl)`**

Definition at line 1089 of file `fipa_envelope.c`.

References `mxmlNewElement()`, `mxmlNewText()`, and `node`.

Referenced by `fipa_envelope_Compose__params()`.

#### **13.34.2.12 `mxml_node_t* fipa_envelope_Compose__date (fipa_acl_message_t * fipa_acl)`**

Definition at line 1117 of file `fipa_envelope.c`.

References `buf`, `mxmlNewElement()`, `mxmlNewText()`, and `node`.

Referenced by `fipa_envelope_Compose__params()`.

#### **13.34.2.13 `mxml_node_t* fipa_envelope_Compose__envelope (fipa_acl_message_t * fipa_acl)`**

Definition at line 894 of file `fipa_envelope.c`.

References `fipa_envelope_Compose__params()`, `MXML_ADD_AFTER`, `MXML_ADD_TO_PARENT`, `mxmlAdd()`, `mxmlNewElement()`, and `node`.

Referenced by `fipa_envelope_Compose()`.

#### **13.34.2.14 `mxml_node_t* fipa_envelope_Compose__from (fipa_acl_message_t * fipa_acl)`**

Definition at line 1021 of file `fipa_envelope.c`.

References `fipa_agent_identifier_s::addresses`, `buf`, `dynstring_Append()`, `dynstring_Destroy()`, `dynstring_New()`, `g_mc_platform`, `mc_platform_s::hostname`, `dynstring_s::message`, `mxmlNewElement()`, `mxmlNewText()`, `fipa_agent_identifier_s::name`, `fipa_url_sequence_s::num`, `mc_platform_s::port`, `fipa_acl_message_s::sender`, `fipa_url_s::str`, and `fipa_url_sequence_s::urls`.

Referenced by `fipa_envelope_Compose__params()`.

### 13.34.2.15 **mxml\_node\_t\* fipa\_envelope\_Compose\_\_intended\_receiver (fipa\_acl\_message\_t \* *fipa\_acl*)**

Definition at line 1146 of file fipa\_envelope.c.

References fipa\_agent\_identifier\_s::addresses, fipa\_agent\_identifier\_set\_s::fipa\_agent\_identifiers, mxml-NewElement(), mxmlNewText(), fipa\_agent\_identifier\_s::name, node, fipa\_url\_sequence\_s::num, fipa\_acl\_message\_s::receiver, fipa\_url\_s::str, and fipa\_url\_sequence\_s::urls.

Referenced by fipa\_envelope\_Compose\_\_params().

### 13.34.2.16 **mxml\_node\_t\* fipa\_envelope\_Compose\_\_params (fipa\_acl\_message\_t \* *fipa\_acl*)**

Definition at line 914 of file fipa\_envelope.c.

References fipa\_envelope\_Compose\_\_acl\_representation(), fipa\_envelope\_Compose\_\_date(), fipa\_envelope\_Compose\_\_from(), fipa\_envelope\_Compose\_\_intended\_receiver(), fipa\_envelope\_Compose\_\_payload\_encoding(), fipa\_envelope\_Compose\_\_to(), MXML\_ADD\_AFTER, MXML\_ADD\_TO\_PARENT, mxmlAdd(), mxmlElementSetAttr(), mxmlNewElement(), and node.

Referenced by fipa\_envelope\_Compose\_\_envelope().

### 13.34.2.17 **mxml\_node\_t\* fipa\_envelope\_Compose\_\_payload\_encoding (fipa\_acl\_message\_t \* *fipa\_acl*)**

Definition at line 1103 of file fipa\_envelope.c.

References mxmlNewElement(), mxmlNewText(), and node.

Referenced by fipa\_envelope\_Compose\_\_params().

### 13.34.2.18 **mxml\_node\_t\* fipa\_envelope\_Compose\_\_to (fipa\_acl\_message\_t \* *fipa\_acl*)**

Definition at line 975 of file fipa\_envelope.c.

References fipa\_agent\_identifier\_s::addresses, fipa\_agent\_identifier\_set\_s::fipa\_agent\_identifiers, mxml-NewElement(), mxmlNewText(), fipa\_agent\_identifier\_s::name, node, fipa\_url\_sequence\_s::num, fipa\_agent\_identifier\_set\_s::num, fipa\_acl\_message\_s::receiver, fipa\_url\_s::str, and fipa\_url\_sequence\_s::urls.

Referenced by fipa\_envelope\_Compose\_\_params().

### 13.34.2.19 **int fipa\_envelope\_HandleAclRepresentation (struct fipa\_acl\_envelope\_s \* *envelope*, mxml\_node\_t \* *param\_node*, int *cur\_param*)**

Definition at line 381 of file fipa\_envelope.c.

References fipa\_acl\_Param\_s::acl\_representation, mxml\_node\_s::child, MC\_ERR\_PARSE, MXML\_DESCEND\_FIRST, MXML\_TEXT, mxmlFindElement(), fipa\_acl\_envelope\_s::params, mxml\_text\_s::string, mxml\_value\_u::text, mxml\_node\_s::type, and mxml\_node\_s::value.

Referenced by fipa\_envelope\_HandleParams().



**13.34.2.20** `int fipa_envelope_HandleComments (struct fipa_acl_envelope_s * envelope,  
mxml_node_t * param_node, int cur_param)`

Definition at line 357 of file fipa\_envelope.c.

References mxml\_node\_s::child, fipa\_acl\_Param\_s::comments, MC\_ERR\_PARSE, MXML\_DESCEND\_FIRST, MXML\_TEXT, mxmlFindElement(), fipa\_acl\_envelope\_s::params, mxml\_text\_s::string, mxml\_value\_u::text, mxml\_node\_s::type, and mxml\_node\_s::value.

Referenced by fipa\_envelope\_HandleParams().

**13.34.2.21** `int fipa_envelope_HandleDate (struct fipa_acl_envelope_s * envelope, mxml_node_t *  
param_node, int cur_param)`

Definition at line 452 of file fipa\_envelope.c.

References mxml\_node\_s::child, fipa\_acl\_Param\_s::date, fipa\_datetime\_Parse(), MC\_ERR\_PARSE, fipa\_message\_string\_s::message, MXML\_DESCEND\_FIRST, MXML\_TEXT, mxmlFindElement(), fipa\_acl\_envelope\_s::params, fipa\_message\_string\_s::parse, mxml\_text\_s::string, mxml\_value\_u::text, mxml\_node\_s::type, and mxml\_node\_s::value.

Referenced by fipa\_envelope\_HandleParams().

**13.34.2.22** `int fipa_envelope_HandleEnvelope (struct fipa_acl_envelope_s * envelope,  
mxml_node_t * node)`

Definition at line 178 of file fipa\_envelope.c.

References fipa\_envelope\_HandleParams(), MC\_ERR\_PARSE, MXML\_DESCEND\_FIRST, and mxmlFindElement().

Referenced by fipa\_envelope\_Parse().

**13.34.2.23** `int fipa_envelope_HandleFrom (struct fipa_acl_envelope_s * envelope, mxml_node_t *  
param_node, int cur_param)`

Definition at line 328 of file fipa\_envelope.c.

References fipa\_envelope\_ParseAgentIdentifier(), fipa\_acl\_Param\_s::from, MC\_ERR\_PARSE, MXML\_DESCEND\_FIRST, mxmlFindElement(), and fipa\_acl\_envelope\_s::params.

Referenced by fipa\_envelope\_HandleParams().

**13.34.2.24** `int fipa_envelope_HandleIntendedReceiver (struct fipa_acl_envelope_s * envelope,  
mxml_node_t * param_node, int cur_param)`

Definition at line 482 of file fipa\_envelope.c.

References fipa\_agent\_identifier\_set\_New(), fipa\_agent\_identifier\_set\_s::fipa\_agent\_identifiers, fipa\_envelope\_ParseAgentIdentifier(), fipa\_acl\_Param\_s::intended\_receiver, MC\_ERR\_PARSE, MXML\_DESCEND\_FIRST, MXML\_NO\_DESCEND, mxmlFindElement(), fipa\_agent\_identifier\_set\_s::num, fipa\_acl\_envelope\_s::params, and fipa\_agent\_identifier\_set\_s::retain\_order.

Referenced by fipa\_envelope\_HandleParams().

### 13.34.2.25 **int fipa\_envelope\_HandleParams** (struct fipa\_acl\_envelope\_s \* *envelope*, mxml\_node\_t \* *node*)

Definition at line 198 of file fipa\_envelope.c.

References `buf`, `fipa_envelope_HandleAclRepresentation()`, `fipa_envelope_HandleComments()`, `fipa_envelope_HandleDate()`, `fipa_envelope_HandleFrom()`, `fipa_envelope_HandleIntendedReceiver()`, `fipa_envelope_HandlePayloadEncoding()`, `fipa_envelope_HandlePayloadLength()`, `fipa_envelope_HandleReceived()`, `fipa_envelope_HandleTo()`, `MC_ERR_PARSE`, `MXML_DESCEND_FIRST`, `mxmFindElement()`, `fipa_acl_envelope_s::num_params`, and `fipa_acl_envelope_s::params`.

Referenced by `fipa_envelope_HandleEnvelope()`.

### 13.34.2.26 **int fipa\_envelope\_HandlePayloadEncoding** (struct fipa\_acl\_envelope\_s \* *envelope*, mxml\_node\_t \* *param\_node*, int *cur\_param*)

Definition at line 428 of file fipa\_envelope.c.

References `mxml_node_s::child`, `MC_ERR_PARSE`, `MXML_DESCEND_FIRST`, `MXML_TEXT`, `mxmFindElement()`, `fipa_acl_envelope_s::params`, `fipa_acl_Param_s::payload_encoding`, `mxml_text_s::string`, `mxml_value_u::text`, `mxml_node_s::type`, and `mxml_node_s::value`.

Referenced by `fipa_envelope_HandleParams()`.

### 13.34.2.27 **int fipa\_envelope\_HandlePayloadLength** (struct fipa\_acl\_envelope\_s \* *envelope*, mxml\_node\_t \* *param\_node*, int *cur\_param*)

Definition at line 405 of file fipa\_envelope.c.

References `mxml_node_s::child`, `MC_ERR_PARSE`, `MXML_DESCEND_FIRST`, `MXML_TEXT`, `mxmFindElement()`, `fipa_acl_envelope_s::params`, `fipa_acl_Param_s::payload_length`, `mxml_text_s::string`, `mxml_value_u::text`, `mxml_node_s::type`, and `mxml_node_s::value`.

Referenced by `fipa_envelope_HandleParams()`.

### 13.34.2.28 **int fipa\_envelope\_HandleReceived** (struct fipa\_acl\_envelope\_s \* *envelope*, mxml\_node\_t \* *param\_node*, int *cur\_param*)

Definition at line 560 of file fipa\_envelope.c.

References `fipa_acl_envelope_Received_New()`, `fipa_datetime_Parse()`, `fipa_url_New()`, `MC_ERR_PARSE`, `fipa_message_string_s::message`, `MXML_DESCEND_FIRST`, `mxmlElementGetAttr()`, `mxmFindElement()`, `node`, `fipa_acl_envelope_s::params`, `fipa_message_string_s::parse`, `fipa_acl_Param_s::received`, `fipa_acl_envelope_Received_s::received_by`, `fipa_acl_envelope_Received_s::received_date`, `fipa_acl_envelope_Received_s::received_from`, `fipa_acl_envelope_Received_s::received_id`, `fipa_acl_envelope_Received_s::received_via`, and `fipa_url_s::str`.

Referenced by `fipa_envelope_HandleParams()`.

### 13.34.2.29 **int fipa\_envelope\_HandleTo** (struct fipa\_acl\_envelope\_s \* *envelope*, mxml\_node\_t \* *param\_node*, int *cur\_param*)

Definition at line 250 of file fipa\_envelope.c.

References `fipa_acl_Param_New()`, `fipa_agent_identifier_set_New()`, `fipa_agent_identifier_set_s::fipa_agent_identifiers`, `fipa_envelope_ParseAgentIdentifier()`, `MC_ERR_PARSE`, `MXML_DESCEND_FIRST`,

MXML\_NO\_DESCEND, mxmlFindElement(), fipa\_agent\_identifier\_set\_s::num, fipa\_acl\_envelope\_s::params, fipa\_agent\_identifier\_set\_s::retain\_order, and fipa\_acl\_Param\_s::to.

Referenced by fipa\_envelope\_HandleParams().

### 13.34.2.30 int fipa\_envelope\_Parse (struct fipa\_acl\_envelope\_s \* *envelope*, const char \* *message*)

Definition at line 165 of file fipa\_envelope.c.

References fipa\_envelope\_HandleEnvelope(), MXML\_NO\_CALLBACK, mxmlDelete(), and mxmlLoadString().

Referenced by acc\_connection\_Thread().

### 13.34.2.31 int fipa\_envelope\_ParseAddresses (struct fipa\_agent\_identifier\_s \* *aid*, mxml\_node\_t \* *addresses\_node*)

Definition at line 737 of file fipa\_envelope.c.

References fipa\_agent\_identifier\_s::addresses, mxml\_node\_s::child, fipa\_url\_New(), MC\_ERR\_PARSE, MXML\_DESCEND\_FIRST, MXML\_NO\_DESCEND, MXML\_TEXT, mxmlFindElement(), fipa\_url\_sequence\_s::num, fipa\_url\_s::str, mxml\_text\_s::string, mxml\_value\_u::text, mxml\_node\_s::type, fipa\_url\_sequence\_s::urls, and mxml\_node\_s::value.

Referenced by fipa\_envelope\_ParseAgentIdentifier().

### 13.34.2.32 int fipa\_envelope\_ParseAgentIdentifier (struct fipa\_agent\_identifier\_s \*\* *aid*, mxml\_node\_t \* *agent\_identifier\_node*)

Definition at line 687 of file fipa\_envelope.c.

References mxml\_node\_s::child, fipa\_agent\_identifier\_New(), fipa\_envelope\_ParseAddresses(), fipa\_envelope\_ParseResolvers(), MC\_ERR\_PARSE, MXML\_DESCEND\_FIRST, MXML\_TEXT, mxmlFindElement(), mxml\_text\_s::string, mxml\_value\_u::text, mxml\_node\_s::type, and mxml\_node\_s::value.

Referenced by fipa\_envelope\_HandleFrom(), fipa\_envelope\_HandleIntendedReceiver(), fipa\_envelope\_HandleTo(), and fipa\_envelope\_ParseResolvers().

### 13.34.2.33 int fipa\_envelope\_ParseResolvers (struct fipa\_agent\_identifier\_s \* *aid*, mxml\_node\_t \* *resolvers\_node*)

Definition at line 804 of file fipa\_envelope.c.

References fipa\_agent\_identifier\_set\_s::fipa\_agent\_identifiers, fipa\_envelope\_ParseAgentIdentifier(), MC\_ERR\_PARSE, MXML\_DESCEND\_FIRST, MXML\_NO\_DESCEND, mxmlFindElement(), fipa\_agent\_identifier\_set\_s::num, fipa\_agent\_identifier\_s::resolvers, and fipa\_agent\_identifier\_set\_s::retain\_order.

Referenced by fipa\_envelope\_ParseAgentIdentifier().

## 13.35 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/fipa\_comm.h File Reference

```
#include <fipa_acl.h>
```

### Data Structures

- struct [fipa\\_comm\\_message\\_check\\_t](#)
- struct [fipa\\_comm\\_reply\\_t](#)
- struct [fipa\\_comm\\_action\\_t](#)
- struct [fipa\\_comm\\_performative\\_t](#)
- struct [fipa\\_list\\_t](#)
- struct [fipa\\_comm\\_protocol\\_cn\\_t](#)
- struct [fipa\\_comm\\_protocol\\_t](#)
- struct [fipa\\_comm\\_t](#)

### Defines

- #define [true](#) 1
- #define [false](#) 0
- #define [MC\\_NUM\\_PERFORMATIVES](#) 23

### Typedefs

- typedef [int](#)(\* [fipa\\_comm\\_callback](#) )(char \*received\_content, char \*\*reply\_content)
- typedef [int](#)(\* [protocol\\_contract\\_net\\_callback](#) )(int numBids, char \*\*bids, char \*winners)
- typedef struct [fipa\\_comm\\_message\\_check\\_t](#) [fipa\\_comm\\_message\\_check\\_s](#)
- typedef struct [fipa\\_comm\\_message\\_check\\_t](#) \* [fipa\\_comm\\_message\\_check\\_p](#)
- typedef struct [fipa\\_comm\\_reply\\_t](#) [fipa\\_comm\\_reply\\_s](#)
- typedef struct [fipa\\_comm\\_reply\\_t](#) \* [fipa\\_comm\\_reply\\_p](#)
- typedef struct [fipa\\_comm\\_action\\_t](#) [fipa\\_comm\\_action\\_s](#)
- typedef struct [fipa\\_comm\\_action\\_t](#) \* [fipa\\_comm\\_action\\_p](#)
- typedef struct [fipa\\_comm\\_performative\\_t](#) [fipa\\_comm\\_performative\\_s](#)
- typedef struct [fipa\\_comm\\_performative\\_t](#) \* [fipa\\_comm\\_performative\\_p](#)
- typedef struct [fipa\\_list\\_t](#) [fipa\\_list\\_s](#)
- typedef struct [fipa\\_list\\_t](#) \* [fipa\\_list\\_p](#)
- typedef struct [fipa\\_comm\\_protocol\\_cn\\_t](#) [fipa\\_comm\\_protocol\\_cn\\_s](#)
- typedef struct [fipa\\_comm\\_protocol\\_cn\\_t](#) \* [fipa\\_comm\\_protocol\\_cn\\_p](#)
- typedef struct [fipa\\_comm\\_protocol\\_t](#) \* [fipa\\_comm\\_protocol\\_p](#)
- typedef struct [fipa\\_comm\\_protocol\\_t](#) [fipa\\_comm\\_protocol\\_s](#)
- typedef struct [fipa\\_comm\\_t](#) [fipa\\_comm\\_s](#)
- typedef struct [fipa\\_comm\\_t](#) \* [fipa\\_comm\\_p](#)
- typedef struct [fipa\\_comm\\_t](#) [MCFIPAComm\\_s](#)
- typedef struct [fipa\\_comm\\_t](#) \* [MCFIPAComm\\_p](#)

## Enumerations

- enum `fipa_comm_type_e` {  
`FIPA_COMM_ACTION`, `FIPA_COMM_CHECK`, `FIPA_COMM_REPLY`, `FIPA_COMM_CALLBACK`,  
`FIPA_COMM_END` }
- enum `fipa_comm_logic_type_e` { `FIPA_COMM_LOGIC_ERROR` = -1, `FIPA_COMM_LOGIC_OR`, `FIPA_COMM_LOGIC_AND` }
- enum `fipa_comm_check_type_e` {  
`FIPA_COMM_CHECK_ERROR` = -1, `FIPA_COMM_CHECK_SENDER_NAME`, `FIPA_COMM_CHECK_SENDER_ADDRESS`, `FIPA_COMM_CHECK_CONTENT`,  
`FIPA_COMM_CHECK_CONVERSATIONID`, `FIPA_COMM_CHECK_LANGUAGE`, `FIPA_COMM_CHECK_ONTOLOGY`, `FIPA_COMM_CHECK_ENCODING` }
- enum `fipa_comm_action_type_e` {  
`FIPA_COMM_ACTION_ERROR` = -1, `FIPA_COMM_ACTION_BOTH`, `FIPA_COMM_ACTION_REPLY`, `FIPA_COMM_ACTION_CALLBACK`,  
`FIPA_COMM_ACTION_NONE` }
- enum `fipa_comm_protocol_cn_state_e` {  
`FIPA_COMM_PROTOCOL_CN_ERROR` = -1, `FIPA_COMM_PROTOCOL_CN_START`, `FIPA_COMM_PROTOCOL_CN_CFP`, `FIPA_COMM_PROTOCOL_CN_BID`,  
`FIPA_COMM_PROTOCOL_CN_RESULT`, `FIPA_COMM_PROTOCOL_CN_DONE` }

## Functions

- `fipa_comm_p mc_FIPAComm_New ()`
- `int mc_FIPAComm_HandleMessage (fipa_comm_p fcomm, fipa_acl_message_p acl)`
- `int mc_FIPAComm_HandleMessageOnce (fipa_comm_p fcomm)`
- `int mc_FIPAComm_HandleMessageWait (fipa_comm_p fcomm)`
- `void mc_FIPAComm_Destroy (fipa_comm_p)`
- `int mc_FIPAComm_RegisterPerformative (fipa_comm_p fcomm, enum fipa_performative_e type)`
- `int mc_FIPAComm_DeregisterPerformative (fipa_comm_p fcomm, enum fipa_performative_e type)`
- `int mc_FIPAComm_SetDefaultReply (fipa_comm_p fcomm, enum fipa_performative_e type, char *content)`
- `int mc_FIPAComm_SetPerformativeDefaultReply (fipa_comm_p fcomm, enum fipa_performative_e ptype, enum fipa_performative_e rtype, char *content)`
- `int mc_FIPAComm_SetAgent (fipa_comm_p fcomm, void *agent, char *agent_name, char *agent_address)`
- `int mc_FIPAComm_RegisterAction (fipa_comm_p fcomm, enum fipa_performative_e pType, enum fipa_comm_action_type_e aType,...)`
- `int mc_FIPAComm_RegisterActionS (fipa_performative_e type, fipa_comm_action_p act)`
- `fipa_comm_protocol_p mc_FIPAComm_New ()`
- `int mc_FIPAComm_RegisterProtocolCallback (fipa_comm_p fcomm, enum fipa_protocol_e protocolType, fipa_comm_callback func)`
- `int mc_FIPAComm_RegisterProtocol (fipa_comm_p fcomm, enum fipa_protocol_e protocolType, char *protocolID, void *info, char *network)`
- `void fipa_comm_protocol_info_COPY (fipa_comm_protocol_p proto, void *info)`
- `int fipa_list_Add (fipa_list_p *list, fipa_acl_message_p acl)`
- `int mc_FIPAComm_StartProtocol (fipa_comm_p fcomm, enum fipa_protocol_e type, char *id)`
- `int mc_FIPAComm_HandleProtocol (fipa_comm_p fcomm, fipa_acl_message_p acl)`

- [fipa\\_comm\\_protocol\\_p mc\\_FIPAComm\\_AddNewProtocol](#) ([fipa\\_comm\\_p](#) fcomm, [enum fipa\\_protocol\\_e](#) type, [char \\*id](#))
- [int mc\\_FIPAComm\\_FSM\\_CN](#) ([fipa\\_comm\\_p](#) fcomm, [fipa\\_acl\\_message\\_p](#) acl)
- [fipa\\_comm\\_protocol\\_p mc\\_FIPAComm\\_NewProtocol](#) ([enum fipa\\_protocol\\_e](#) type, [char \\*id](#))
- [int mc\\_FIPAComm\\_SetProtocolState](#) ([fipa\\_comm\\_protocol\\_p](#) protocol, [char](#) state)

### 13.35.1 Define Documentation

#### 13.35.1.1 `#define false 0`

Definition at line 11 of file [fipa\\_comm.h](#).

#### 13.35.1.2 `#define MC_NUM_PERFORMATIVES 23`

Definition at line 14 of file [fipa\\_comm.h](#).

#### 13.35.1.3 `#define true 1`

Definition at line 7 of file [fipa\\_comm.h](#).

### 13.35.2 Typedef Documentation

#### 13.35.2.1 `typedef struct fipa_comm_action_t* fipa_comm_action_p`

Definition at line 109 of file [fipa\\_comm.h](#).

#### 13.35.2.2 `typedef struct fipa_comm_action_t fipa_comm_action_s`

Definition at line 108 of file [fipa\\_comm.h](#).

#### 13.35.2.3 `typedef int(* fipa_comm_callback)(char *received_content, char **reply_content)`

Definition at line 17 of file [fipa\\_comm.h](#).

#### 13.35.2.4 `typedef struct fipa_comm_message_check_t* fipa_comm_message_check_p`

Definition at line 80 of file [fipa\\_comm.h](#).

#### 13.35.2.5 `typedef struct fipa_comm_message_check_t fipa_comm_message_check_s`

Definition at line 79 of file [fipa\\_comm.h](#).

#### 13.35.2.6 `typedef struct fipa_comm_t* fipa_comm_p`

Definition at line 193 of file [fipa\\_comm.h](#).

**13.35.2.7 typedef struct fipa\_comm\_performative\_t\* fipa\_comm\_performative\_p**

Definition at line 121 of file fipa\_comm.h.

**13.35.2.8 typedef struct fipa\_comm\_performative\_t fipa\_comm\_performative\_s**

Definition at line 120 of file fipa\_comm.h.

**13.35.2.9 typedef struct fipa\_comm\_protocol\_cn\_t\* fipa\_comm\_protocol\_cn\_p**

Definition at line 156 of file fipa\_comm.h.

**13.35.2.10 typedef struct fipa\_comm\_protocol\_cn\_t fipa\_comm\_protocol\_cn\_s**

Definition at line 155 of file fipa\_comm.h.

**13.35.2.11 typedef struct fipa\_comm\_protocol\_t\* fipa\_comm\_protocol\_p**

Definition at line 172 of file fipa\_comm.h.

**13.35.2.12 typedef struct fipa\_comm\_protocol\_t fipa\_comm\_protocol\_s**

Definition at line 173 of file fipa\_comm.h.

**13.35.2.13 typedef struct fipa\_comm\_reply\_t\* fipa\_comm\_reply\_p**

Definition at line 93 of file fipa\_comm.h.

**13.35.2.14 typedef struct fipa\_comm\_reply\_t fipa\_comm\_reply\_s**

Definition at line 92 of file fipa\_comm.h.

**13.35.2.15 typedef struct fipa\_comm\_t fipa\_comm\_s**

Definition at line 192 of file fipa\_comm.h.

**13.35.2.16 typedef struct fipa\_list\_t\* fipa\_list\_p**

Definition at line 141 of file fipa\_comm.h.

**13.35.2.17 typedef struct fipa\_list\_t fipa\_list\_s**

Definition at line 140 of file fipa\_comm.h.

**13.35.2.18 typedef struct fipa\_comm\_t\* MCFIPAComm\_p**

Definition at line 195 of file fipa\_comm.h.

**13.35.2.19 typedef struct fipa\_comm\_t MCFIPAComm\_s**

Definition at line 194 of file fipa\_comm.h.

**13.35.2.20 typedef int(\* protocol\_contract\_net\_callback)(int numBids, char \*\*bids, char \*winners)**

Definition at line 18 of file fipa\_comm.h.

**13.35.3 Enumeration Type Documentation****13.35.3.1 enum fipa\_comm\_action\_type\_e**

Enumerator:

*FIPA\_COMM\_ACTION\_ERROR*  
*FIPA\_COMM\_ACTION\_BOTH*  
*FIPA\_COMM\_ACTION\_REPLY*  
*FIPA\_COMM\_ACTION\_CALLBACK*  
*FIPA\_COMM\_ACTION\_NONE*

Definition at line 45 of file fipa\_comm.h.

**13.35.3.2 enum fipa\_comm\_check\_type\_e**

Enumerator:

*FIPA\_COMM\_CHECK\_ERROR*  
*FIPA\_COMM\_CHECK\_SENDER\_NAME*  
*FIPA\_COMM\_CHECK\_SENDER\_ADDRESS*  
*FIPA\_COMM\_CHECK\_CONTENT*  
*FIPA\_COMM\_CHECK\_CONVERSATIONID*  
*FIPA\_COMM\_CHECK\_LANGUAGE*  
*FIPA\_COMM\_CHECK\_ONTOLOGY*  
*FIPA\_COMM\_CHECK\_ENCODING*

Definition at line 34 of file fipa\_comm.h.

**13.35.3.3 enum fipa\_comm\_logic\_type\_e**

Enumerator:

*FIPA\_COMM\_LOGIC\_ERROR*  
*FIPA\_COMM\_LOGIC\_OR*



### *FIPA\_COMM\_LOGIC\_AND*

Definition at line 28 of file fipa\_comm.h.

#### 13.35.3.4 enum fipa\_comm\_protocol\_cn\_state\_e

Enumerator:

*FIPA\_COMM\_PROTOCOL\_CN\_ERROR*

*FIPA\_COMM\_PROTOCOL\_CN\_START*

*FIPA\_COMM\_PROTOCOL\_CN\_CFP*

*FIPA\_COMM\_PROTOCOL\_CN\_BID*

*FIPA\_COMM\_PROTOCOL\_CN\_RESULT*

*FIPA\_COMM\_PROTOCOL\_CN\_DONE*

Definition at line 127 of file fipa\_comm.h.

#### 13.35.3.5 enum fipa\_comm\_type\_e

Enumerator:

*FIPA\_COMM\_ACTION*

*FIPA\_COMM\_CHECK*

*FIPA\_COMM\_REPLY*

*FIPA\_COMM\_CALLBACK*

*FIPA\_COMM\_END*

Definition at line 20 of file fipa\_comm.h.



### 13.35.4 Function Documentation

- 13.35.4.1 void fipa\_comm\_protocol\_info\_COPY (fipa\_comm\_protocol\_p *proto*, void \* *info*)
- 13.35.4.2 int fipa\_list\_Add (fipa\_list\_p \* *list*, fipa\_acl\_message\_p *acl*)
- 13.35.4.3 fipa\_comm\_protocol\_p mc\_FIPAComm\_AddNewProtocol (fipa\_comm\_p *fcomm*, enum fipa\_protocol\_e *type*, char \* *id*)
- 13.35.4.4 int mc\_FIPAComm\_DeregisterPerformative (fipa\_comm\_p *fcomm*, enum fipa\_performative\_e *type*)
- 13.35.4.5 void mc\_FIPAComm\_Destroy (fipa\_comm\_p)
- 13.35.4.6 int mc\_FIPAComm\_FSM\_CN (fipa\_comm\_p *fcomm*, fipa\_acl\_message\_p *acl*)
- 13.35.4.7 int mc\_FIPAComm\_HandleMessage (fipa\_comm\_p *fcomm*, fipa\_acl\_message\_p *acl*)
- 13.35.4.8 int mc\_FIPAComm\_HandleMessageOnce (fipa\_comm\_p *fcomm*)
- 13.35.4.9 int mc\_FIPAComm\_HandleMessageWait (fipa\_comm\_p *fcomm*)
- 13.35.4.10 int mc\_FIPAComm\_HandleProtocol (fipa\_comm\_p *fcomm*, fipa\_acl\_message\_p *acl*)
- 13.35.4.11 fipa\_comm\_p mc\_FIPAComm\_New ()
- 13.35.4.12 fipa\_comm\_protocol\_p mc\_FIPAComm\_NewProtocol (enum fipa\_protocol\_e *type*, char \* *id*)
- 13.35.4.13 int mc\_FIPAComm\_RegisterAction (fipa\_comm\_p *fcomm*, enum fipa\_performative\_e *pType*, enum fipa\_comm\_action\_type\_e *aType*, ...)
- 13.35.4.14 int mc\_FIPAComm\_RegisterActionS (fipa\_performative\_e *type*, fipa\_comm\_action\_p *act*)
- 13.35.4.15 int mc\_FIPAComm\_RegisterPerformative (fipa\_comm\_p *fcomm*, enum fipa\_performative\_e *type*)
- 13.35.4.16 int mc\_FIPAComm\_RegisterProtocol (fipa\_comm\_p *fcomm*, enum fipa\_protocol\_e *protocolType*, char \* *protocolID*, void \* *info*, char \* *network*)
- 13.35.4.17 int mc\_FIPAComm\_RegisterProtocolCallback (fipa\_comm\_p *fcomm*, enum fipa\_protocol\_e *protocolType*, fipa\_comm\_callback\_func)
- 13.35.4.18 int mc\_FIPAComm\_SetAgent (fipa\_comm\_p *fcomm*, void \* *agent*, char \* *agent\_name*, char \* *agent\_address*)
- 13.35.4.19 int mc\_FIPAComm\_SetDefaultReply (fipa\_comm\_p *fcomm*, enum fipa\_performative\_e *type*, char \* *content*)
- 13.35.4.20 int mc\_FIPAComm\_SetPerformativeDefaultReply (fipa\_comm\_p *fcomm*, enum fipa\_performative\_e *ptype*, enum fipa\_performative\_e *rtype*, char \* *content*)
- 13.35.4.21 int mc\_FIPAComm\_SetProtocolState (fipa\_comm\_protocol\_p *protocol*, char *state*)
- 13.35.4.22 int mc\_FIPAComm\_StartProtocol (fipa\_comm\_p *fcomm*, enum fipa\_protocol\_e *type*, char \* *id*)
- 13.35.4.23 fipa\_comm\_protocol\_p mc\_FIPAProtocol\_New ()

## 13.36 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/host\_id.h File Reference

### Data Structures

- struct [host\\_id\\_s](#)

### Typedefs

- typedef struct [host\\_id\\_s](#) [host\\_id\\_t](#)
- typedef [host\\_id\\_t](#) \* [host\\_id\\_p](#)

### Functions

- [host\\_id\\_p](#) [host\\_id\\_Initialize](#) (const char \*hostname, [int](#) port)
- void [host\\_id\\_Destroy](#) ([host\\_id\\_p](#) host\_id)
- [int](#) [host\\_id\\_Compare](#) ([host\\_id\\_p](#) host1, [host\\_id\\_p](#) host2)

#### 13.36.1 Typedef Documentation

##### 13.36.1.1 typedef [host\\_id\\_t](#)\* [host\\_id\\_p](#)

Definition at line 43 of file [host\\_id.h](#).

##### 13.36.1.2 typedef struct [host\\_id\\_s](#) [host\\_id\\_t](#)

#### 13.36.2 Function Documentation

##### 13.36.2.1 [int](#) [host\\_id\\_Compare](#) ([host\\_id\\_p](#) *host1*, [host\\_id\\_p](#) *host2*)

##### 13.36.2.2 void [host\\_id\\_Destroy](#) ([host\\_id\\_p](#) *host\_id*)

##### 13.36.2.3 [host\\_id\\_p](#) [host\\_id\\_Initialize](#) (const char \* *hostname*, [int](#) *port*)

## 13.37 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/interpreter\_variable\_data.h File Reference

```
#include <ch.h>
```

### Data Structures

- struct [interpreter\\_variable\\_data\\_s](#)

### Typedefs

- typedef struct [interpreter\\_variable\\_data\\_s](#) [interpreter\\_variable\\_data\\_t](#)
- typedef [interpreter\\_variable\\_data\\_t](#) \* [interpreter\\_variable\\_data\\_p](#)

### Functions

- [interpreter\\_variable\\_data\\_p](#) [interpreter\\_variable\\_data\\_New](#) (void)
- [interpreter\\_variable\\_data\\_p](#) [interpreter\\_variable\\_data\\_Copy](#) ([interpreter\\_variable\\_data\\_p](#) src)
- [interpreter\\_variable\\_data\\_p](#) [interpreter\\_variable\\_data\\_InitializeFromAgent](#) (struct [agent\\_s](#) \*agent)
- [interpreter\\_variable\\_data\\_p](#) [interpreter\\_variable\\_data\\_Initialize](#) (struct [agent\\_s](#) \*agent, const char \*varname)
- int [interpreter\\_variable\\_data\\_Destroy](#) ([interpreter\\_variable\\_data\\_p](#) agent\_return\_data)

### 13.37.1 Typedef Documentation

#### 13.37.1.1 typedef [interpreter\\_variable\\_data\\_t](#)\* [interpreter\\_variable\\_data\\_p](#)

Definition at line 51 of file [interpreter\\_variable\\_data.h](#).

#### 13.37.1.2 typedef struct [interpreter\\_variable\\_data\\_s](#) [interpreter\\_variable\\_data\\_t](#)

Definition at line 50 of file [interpreter\\_variable\\_data.h](#).

### 13.37.2 Function Documentation

#### 13.37.2.1 [interpreter\\_variable\\_data\\_p](#) [interpreter\\_variable\\_data\\_Copy](#) ([interpreter\\_variable\\_data\\_p](#) src)

Definition at line 235 of file [agent\\_return\\_data.c](#).

References [interpreter\\_variable\\_data\\_s::array\\_dim](#), [interpreter\\_variable\\_data\\_s::array\\_extent](#), [interpreter\\_variable\\_data\\_s::data](#), [interpreter\\_variable\\_data\\_s::data\\_type](#), [interpreter\\_variable\\_data\\_InitializeFromAgent](#), [interpreter\\_variable\\_data\\_Initialize](#), [interpreter\\_variable\\_data\\_Destroy](#), [interpreter\\_variable\\_data\\_s::name](#), and [interpreter\\_variable\\_data\\_s::size](#).

Referenced by [agent\\_task\\_Copy](#)().

**13.37.2.2 int interpreter\_variable\_data\_Destroy (interpreter\_variable\_data\_p agent\_return\_data)**

Definition at line 216 of file agent\_return\_data.c.

References interpreter\_variable\_data\_s::array\_extent, interpreter\_variable\_data\_s::data, MC\_SUCCESS, and interpreter\_variable\_data\_s::name.

Referenced by agent\_RunChScriptThread(), and agent\_task\_Destroy().

**13.37.2.3 interpreter\_variable\_data\_p interpreter\_variable\_data\_Initialize (struct agent\_s \* agent, const char \* varname)****13.37.2.4 interpreter\_variable\_data\_p interpreter\_variable\_data\_InitializeFromAgent (struct agent\_s \* agent)****13.37.2.5 interpreter\_variable\_data\_p interpreter\_variable\_data\_New (void)**

Definition at line 46 of file agent\_return\_data.c.

References interpreter\_variable\_data\_s::array\_dim, interpreter\_variable\_data\_s::array\_extent, CHECK\_NULL, interpreter\_variable\_data\_s::data, interpreter\_variable\_data\_s::data\_type, interpreter\_variable\_data\_s::name, and interpreter\_variable\_data\_s::size.

Referenced by agent\_xml\_parse\_\_data(), interpreter\_variable\_data\_Copy(), and MC\_SaveData\_chdl().

## 13.38 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/libmc.h File Reference

MobileC api header file. `#include <embedch.h>`

### Data Structures

- struct [agency\\_s](#)  
*The agency handle.*
- struct [MCAgencyOptions\\_s](#)  
*User modifiable agency options.*
- struct [agent\\_thread\\_arg\\_s](#)

### Defines

- `#define` [THREAD\\_T](#) `pthread_t`
- `#define` [EXPORTMC](#)
- `#define` [MC\\_Wait](#)(arg1) `MC_MainLoop(arg1)`

### Typedefs

- typedef enum [error\\_code\\_e](#) `error_code_t`
- typedef enum [MC\\_SteerCommand\\_e](#) `MC_SteerCommand_t`  
*Available commands for MC\_Steer.*
- typedef struct [agency\\_s](#) `agency_t`  
*The agency handle.*
- typedef [agency\\_t](#) \* `agency_p`
- typedef [agency\\_p](#) `MCAgency_t`
- typedef struct [MCAgencyOptions\\_s](#) `MCAgencyOptions_t`  
*User modifiable agency options.*
- typedef struct [agent\\_thread\\_arg\\_s](#) `agent_thread_arg_t`
- typedef struct [agent\\_s](#) `agent_t`
- typedef [agent\\_t](#) \* `MCAgent_t`
- typedef [agent\\_t](#) \* `agent_p`

### Enumerations

- enum [error\\_code\\_e](#) {  
[MC\\_SUCCESS](#) = 0, [MC\\_ERR](#), [MC\\_ERR\\_CONNECT](#), [MC\\_ERR\\_PARSE](#),  
[MC\\_ERR\\_EMPTY](#), [MC\\_ERR\\_INVALID](#), [MC\\_ERR\\_INVALID\\_ARGS](#), [MC\\_ERR\\_NOT\\_FOUND](#),  
[MC\\_ERR\\_MEMORY](#), [MC\\_ERR\\_SEND](#), [MC\\_WARN\\_DUPLICATE](#), [MC\\_SUCCESS](#) = 0,

```
ERR, MC_ERR_CONNECT, MC_ERR_PARSE, MC_ERR_EMPTY,
MC_ERR_INVALID, MC_ERR_INVALID_ARGS, MC_ERR_NOT_FOUND, MC_ERR_
MEMORY,
MC_ERR_SEND, MC_WARN_DUPLICATE }
```

- enum MC\_ThreadIndex\_e {  
MC\_THREAD\_DF = 0, MC\_THREAD\_AMS, MC\_THREAD\_ACC, MC\_THREAD\_CP,  
MC\_THREAD\_AGENT, MC\_THREAD\_ALL }

*MobileC Module indices.*

- enum MC\_SteerCommand\_e { MC\_RUN = 0, MC\_SUSPEND, MC\_RESTART, MC\_STOP }

*Available commands for MC\_Steer.*

- enum MC\_Signal\_e {  
MC\_NO\_SIGNAL = 0x00, MC\_RECV\_CONNECTION = 0x01, MC\_RECV\_MESSAGE = 0x02,  
MC\_RECV\_AGENT = 0x04,  
MC\_RECV\_RETURN = 0x08, MC\_EXEC\_AGENT = 0x10, MC\_ALL\_SIGNALS = 0x20 }

*MobileC system signals.*

- enum MC\_AgentType\_e { MC\_NONE = -1, MC\_REMOTE\_AGENT = 0, MC\_LOCAL\_AGENT,  
MC\_RETURN\_AGENT }
- enum MC\_AgentStatus\_e {  
MC\_NO\_STATUS = -1, MC\_WAIT\_CH = 0, MC\_WAIT\_MESSGSEND, MC\_AGENT\_ACTIVE,  
MC\_AGENT\_NEUTRAL, MC\_AGENT\_SUSPENDED, MC\_WAIT\_FINISHED }

*An agent's current execution state.*

- enum fipa\_performative\_e {  
FIPA\_ERROR = -1, FIPA\_ZERO, FIPA\_ACCEPT\_PROPOSAL, FIPA\_AGREE,  
FIPA\_CANCEL, FIPA\_CALL\_FOR\_PROPOSAL, FIPA\_CONFIRM, FIPA\_DISCONFIRM,  
FIPA\_FAILURE, FIPA\_INFORM, FIPA\_INFORM\_IF, FIPA\_INFORM\_REF,  
FIPA\_NOT\_UNDERSTOOD, FIPA\_PROPOGATE, FIPA\_PROPOSE, FIPA\_PROXY,  
FIPA\_QUERY\_IF, FIPA\_QUERY\_REF, FIPA\_REFUSE, FIPA\_REJECT\_PROPOSAL,  
FIPA\_REQUEST, FIPA\_REQUEST\_WHEN, FIPA\_REQUEST\_WHENEVER, FIPA\_
SUBSCRIBE,  
FIPA\_ERROR = -1, FIPA\_ZERO, FIPA\_ACCEPT\_PROPOSAL, FIPA\_AGREE,  
FIPA\_CANCEL, FIPA\_CALL\_FOR\_PROPOSAL, FIPA\_CONFIRM, FIPA\_DISCONFIRM,  
FIPA\_FAILURE, FIPA\_INFORM, FIPA\_INFORM\_IF, FIPA\_INFORM\_REF,  
FIPA\_NOT\_UNDERSTOOD, FIPA\_PROPOGATE, FIPA\_PROPOSE, FIPA\_PROXY,  
FIPA\_QUERY\_IF, FIPA\_QUERY\_REF, FIPA\_REFUSE, FIPA\_REJECT\_PROPOSAL,  
FIPA\_REQUEST, FIPA\_REQUEST\_WHEN, FIPA\_REQUEST\_WHENEVER, FIPA\_
SUBSCRIBE }



## Functions

- EXPORTMC [int MC\\_AclDestroy](#) (struct [fipa\\_acl\\_message\\_s](#) \*message)  
*Destroy a FIPA ACL message.*
- EXPORTMC struct [fipa\\_acl\\_message\\_s](#) \* [MC\\_AclNew](#) (void)  
*Allocate a new ACL Message.*
- EXPORTMC [int MC\\_AclPost](#) ([MCAgent\\_t](#) agent, struct [fipa\\_acl\\_message\\_s](#) \*message)  
*Post ACL message to agent.*
- EXPORTMC struct [fipa\\_acl\\_message\\_s](#) \* [MC\\_AclReply](#) (struct [fipa\\_acl\\_message\\_s](#) \*acl\_message)  
*Reply to an ACL message.*
- EXPORTMC struct [fipa\\_acl\\_message\\_s](#) \* [MC\\_AclRetrieve](#) ([MCAgent\\_t](#) agent)  
*Retrieve an ACL message.*
- EXPORTMC [int MC\\_AclSend](#) ([MCAgency\\_t](#) attr, struct [fipa\\_acl\\_message\\_s](#) \*acl)  
*Send a composed ACL Message.*
- EXPORTMC struct [fipa\\_acl\\_message\\_s](#) \* [MC\\_AclWaitRetrieve](#) ([MCAgent\\_t](#) agent)  
*Wait for and retrieve an ACL message.*
- EXPORTMC [int MC\\_AclSetProtocol](#) (struct [fipa\\_acl\\_message\\_s](#) \*acl, enum [fipa\\_protocol\\_e](#) performative)
- EXPORTMC [int MC\\_AclSetConversationID](#) (struct [fipa\\_acl\\_message\\_s](#) \*acl, char \*id)
- EXPORTMC [int MC\\_AclSetPerformative](#) (struct [fipa\\_acl\\_message\\_s](#) \*acl, enum [fipa\\_performative\\_e](#) performative)
- EXPORTMC [int MC\\_AclSetSender](#) (struct [fipa\\_acl\\_message\\_s](#) \*acl, const char \*name, const char \*address)
- EXPORTMC [int MC\\_AclAddReceiver](#) (struct [fipa\\_acl\\_message\\_s](#) \*acl, const char \*name, const char \*address)
- EXPORTMC [int MC\\_AclAddReplyTo](#) (struct [fipa\\_acl\\_message\\_s](#) \*acl, const char \*name, const char \*address)
- EXPORTMC [int MC\\_AclSetContent](#) (struct [fipa\\_acl\\_message\\_s](#) \*acl, const char \*content)
- EXPORTMC [int MC\\_AddAgent](#) ([MCAgency\\_t](#) attr, [MCAgent\\_t](#) agent)  
*Add an agent to the agency 'attr'.*
- [int MC\\_AddStationaryAgent](#) ([MCAgency\\_t](#) agency, void \*(\*agent\_thread)(struct [agent\\_thread\\_arg\\_s](#) \*), const char \*name, void \*agent\_args)
- const void \* [MC\\_AgentVariableRetrieve](#) ([MCAgent\\_t](#) agent, const char \*var\_name, [int](#) task\_num)  
*Retrieve a pointer to a previously saved variable.*
- [int MC\\_AgentVariableRetrieveInfo](#) ([MCAgent\\_t](#) agent, const char \*var\_name, [int](#) task\_num, const void \*\*data, [int](#) \*dim, const [int](#) \*\*extent)  
*Retrieve a info about a previously saved variable.*
- [int MC\\_AgentVariableSave](#) ([MCAgent\\_t](#) agent, const char \*var\_name)  
*Mark an agent variable for saving.*

- EXPORTMC [int MC\\_BarrierDelete](#) (MCAgency\_t attr, [int](#) id)  
*Find and delete an initialized MobileC Barrier.*
- EXPORTMC [int MC\\_BarrierInit](#) (MCAgency\_t attr, [int](#) id, [int](#) num\_procs)  
*Initialize a MobileC Barrier.*
- EXPORTMC [int MC\\_CallAgentFunc](#) (MCAgent\_t agent, const char \*funcName, void \*returnVal, [int](#) numArgs,...)  
*Use custom ChOptions\_t type for internal Ch interpreter.*
- EXPORTMC [int MC\\_CallAgentFuncArg](#) (MCAgent\_t agent, const char \*funcName, void \*returnVal, void \*arg)  
*Calls a function defined in an agent.*
- EXPORTMC [int MC\\_CallAgentFuncV](#) (MCAgent\_t agent, const char \*funcName, void \*returnVal, va\_list ap)  
*Calls a function defined in an agent.*
- EXPORTMC [int MC\\_CallAgentFuncVar](#) (MCAgent\_t agent, const char \*funcName, void \*returnVal, ChVaList\_t arglist)
- EXPORTMC [MCAgent\\_t MC\\_ComposeAgent](#) (const char \*name, const char \*home, const char \*owner, const char \*code, const char \*return\_var\_name, const char \*server, [int](#) persistent)  
*Compose a new agent dynamically without using a prewritten XML file.*
- EXPORTMC [MCAgent\\_t MC\\_ComposeAgentS](#) (const char \*name, const char \*home, const char \*owner, const char \*code, const char \*return\_var\_name, const char \*server, const char \*workgroup\_code, [int](#) persistent)  
*Compose a new agent dynamically without using a prewritten XML file.*
- EXPORTMC [MCAgent\\_t MC\\_ComposeAgentFromFile](#) (const char \*name, const char \*home, const char \*owner, const char \*filename, const char \*return\_var\_name, const char \*server, [int](#) persistent)  
*Compose a new agent dynamically from a source code file.*
- EXPORTMC [MCAgent\\_t MC\\_ComposeAgentFromFileS](#) (const char \*name, const char \*home, const char \*owner, const char \*filename, const char \*return\_var\_name, const char \*server, const char \*workgroup\_code, [int](#) persistent)  
*Compose a new agent dynamically from a source code file.*
- EXPORTMC [int MC\\_CondBroadcast](#) (MCAgency\_t attr, [int](#) id)  
*Wakes up all agents/threads waiting on a condition variable.*
- EXPORTMC [int MC\\_CondSignal](#) (MCAgency\_t attr, [int](#) id)  
*Wakes up at least one thread waiting on a condition variable.*
- EXPORTMC [int MC\\_CondReset](#) (MCAgency\_t attr, [int](#) id)  
*Reset a previously signalled MobileC condition variable.*
- EXPORTMC [int MC\\_CondWait](#) (MCAgency\_t attr, [int](#) id)  
*Wait on a MobileC synchronization variable.*
- [int MC\\_CopyAgent](#) (MCAgent\_t \*agent\_out, const [MCAgent\\_t](#) agent\_in)

*Performs a deep-copy of an agent structure.*

- EXPORTMC [int MC\\_DeleteAgent](#) ([MCAgent\\_t](#) agent)  
*Stop and remove an agent.*
- EXPORTMC [int MC\\_DeleteAgentWG](#) ([MCAgent\\_t](#) calling\_agent, [MCAgent\\_t](#) agent)  
*Stop and remove an agent in the same workgroup.*
- EXPORTMC [int MC\\_End](#) ([MCAgency\\_t](#) attr)  
*End an agency.*
- [int MC\\_DestroyServiceSearchResult](#) (char \*\*agentName, char \*\*serviceName, [int](#) \*agentID, [int](#) numResult)  
*Free memory allocated by a Service Search operation.*
- EXPORTMC [MCAgent\\_t MC\\_FindAgentByName](#) ([MCAgency\\_t](#) attr, const char \*name)  
*Find an agent by its name.*
- EXPORTMC [MCAgent\\_t MC\\_FindAgentByID](#) ([MCAgency\\_t](#) attr, [int](#) ID)  
*Find an agent by its id.*
- EXPORTMC void \* [MC\\_GetAgentExecEngine](#) ([MCAgent\\_t](#) agent)  
*Retrieve an agent's Ch interpreter.*
- EXPORTMC [int MC\\_GetAgentID](#) ([MCAgent\\_t](#) agent)  
*Retrieve an agent's id.*
- EXPORTMC char \* [MC\\_GetAgentName](#) ([MCAgent\\_t](#) agent)
- EXPORTMC [int MC\\_GetAgentNumTasks](#) ([MCAgent\\_t](#) agent)  
*Retrive the number of tasks an agent has.*
- EXPORTMC [int MC\\_GetAgentReturnData](#) ([MCAgent\\_t](#) agent, [int](#) task\_num, void \*\*data, [int](#) \*dim, [int](#) \*\*extent)  
*Get an agent's return data.*
- EXPORTMC [int MC\\_GetAgentStatus](#) ([MCAgent\\_t](#) agent)  
*Get an agent's current status.*
- EXPORTMC enum [MC\\_AgentType\\_e MC\\_GetAgentType](#) ([MCAgent\\_t](#) agent)  
*Get an agent's type.*
- EXPORTMC char \* [MC\\_GetAgentXMLString](#) ([MCAgent\\_t](#) agent)  
*Get an agent's xml string.*
- EXPORTMC [int MC\\_HaltAgency](#) ([MCAgency\\_t](#) agency)  
*Halt an agency: Do not process new entries in queues.*
- EXPORTMC [MCAgency\\_t MC\\_Initialize](#) ([int](#) port, [MCAgencyOptions\\_t](#) \*options)  
*Initialize and start a MobileC agency.*

- EXPORTMC [int MC\\_InitializeAgencyOptions](#) (struct [MCAgencyOptions\\_s](#) \*options)  
*Initialize MobileC options.*
- EXPORTMC [int MC\\_MainLoop](#) ([MCAgency\\_t](#) attr)  
*Wait indefinitely.*
- EXPORTMC [int MC\\_LoadAgentFromFile](#) ([MCAgency\\_t](#) attr, const char \*filename)  
*Load an agent from a file into an agency.*
- EXPORTMC [int MC\\_MigrateAgent](#) ([MCAgent\\_t](#) agent, const char \*hostname, [int](#) port)  
*Migrates a running agent to another host.*
- EXPORTMC [int MC\\_MutexLock](#) ([MCAgency\\_t](#) attr, [int](#) id)  
*Locks a MobileC synchronization variable as a mutex.*
- EXPORTMC [int MC\\_MutexUnlock](#) ([MCAgency\\_t](#) attr, [int](#) id)
- EXPORTMC [int MC\\_RegisterService](#) ([MCAgency\\_t](#) agency, [MCAgent\\_t](#) agent, [int](#) agentID, const char \*agentName, char \*\*serviceNames, [int](#) numServices)  
*Register a new service with the Directory Facilitator.*
- EXPORTMC [int MC\\_ResumeAgency](#) ([MCAgency\\_t](#) agency)  
*Resumes a halted agency.*
- EXPORTMC [MCAgent\\_t MC\\_RetrieveAgent](#) ([MCAgency\\_t](#) attr)  
*Retrieves the oldest agent from an agency.*
- EXPORTMC [int MC\\_SemaphorePost](#) ([MCAgency\\_t](#) attr, [int](#) id)  
*Post to a MobileC synchronization variable semaphore.*
- EXPORTMC [int MC\\_SemaphoreWait](#) ([MCAgency\\_t](#) attr, [int](#) id)  
*Decreases a MobileC synchronization variable semaphore count by one.*
- EXPORTMC [int MC\\_SetDefaultAgentStatus](#) ([MCAgency\\_t](#) agency, enum [MC\\_AgentStatus\\_e](#) status)  
*Sets default incoming agent status.*
- EXPORTMC [int MC\\_SetThreadOn](#) ([MCAgencyOptions\\_t](#) \*options, enum [MC\\_ThreadIndex\\_e](#) index)  
*Sets a MobileC thread to "on" status.*
- EXPORTMC [int MC\\_SetThreadsAllOn](#) ([MCAgencyOptions\\_t](#) \*options)  
*Set all Mobile-C threads on.*
- EXPORTMC [int MC\\_SetThreadOff](#) ([MCAgencyOptions\\_t](#) \*options, enum [MC\\_ThreadIndex\\_e](#) index)  
*Sets a MobileC thread to "off" status.*
- EXPORTMC [int MC\\_SetThreadsAllOff](#) ([MCAgencyOptions\\_t](#) \*options)  
*Set all MobileC threads to 'off' status.*

- EXPORTMC [int MC\\_PrintAgentCode](#) ([MCAgent\\_t](#) agent)  
*Prints an agents code to stdout.*
- EXPORTMC [char \\* MC\\_RetrieveAgentCode](#) ([MCAgent\\_t](#) agent)  
*Retrieves an agent's Ch code.*
- EXPORTMC [int MC\\_ResetSignal](#) ([MCAgency\\_t](#) attr)  
*Reset a MobileC signal.*
- EXPORTMC [int MC\\_SearchForService](#) ([MCAgency\\_t](#) attr, [const char \\*searchString](#), [char \\*\\*\\*agentNames](#), [char \\*\\*\\*serviceNames](#), [int \\*\\*agentIDs](#), [int \\*numResults](#))  
*Search the directory facilitator for a service.*
- EXPORTMC [int MC\\_SendAgentMigrationMessage](#) ([MCAgency\\_t](#) attr, [const char \\*message](#), [const char \\*hostname](#), [int port](#))  
*Sends an agent migration message.*
- EXPORTMC [int MC\\_SendAgentMigrationMessageFile](#) ([MCAgency\\_t](#) attr, [const char \\*filename](#), [const char \\*hostname](#), [int port](#))  
*Sends an agent migration message.*
- EXPORTMC [int MC\\_SetAgentStatus](#) ([MCAgent\\_t](#) agent, [int status](#))  
*Set an agent's status.*
- EXPORTMC [int MC\\_Steer](#) ([MCAgency\\_t](#) attr, [int\(\\*funcptr\)\(void \\*data\)](#), [void \\*arg](#))  
*Set up a steerable algorithm.*
- EXPORTMC [enum MC\\_SteerCommand\\_e MC\\_SteerControl](#) ([void](#))  
*The MobileC user-algorithm steering function.*
- EXPORTMC [int MC\\_SyncDelete](#) ([MCAgency\\_t](#) attr, [int id](#))  
*Deletes a previously initialized synchronization variable.*
- EXPORTMC [int MC\\_SyncInit](#) ([MCAgency\\_t](#) attr, [int id](#))  
*Initializes a new MobileC synchronization variable.*
- EXPORTMC [int MC\\_TerminateAgent](#) ([MCAgent\\_t](#) agent)  
*Halt a running agent.*
- EXPORTMC [int MC\\_TerminateAgentWG](#) ([MCAgent\\_t](#) calling\_agent, [MCAgent\\_t](#) agent)
- EXPORTMC [int MC\\_WaitAgent](#) ([MCAgency\\_t](#) attr)  
*Wait indefinitely.*
- EXPORTMC [MCAgent\\_t MC\\_WaitRetrieveAgent](#) ([MCAgency\\_t](#) attr)  
*Wait and retrieve an agent.*
- EXPORTMC [int MC\\_WaitSignal](#) ([MCAgency\\_t](#) attr, [int signals](#))  
*Wait for a MobileC signal.*

### 13.38.1 Detailed Description

MobileC api header file.

Definition in file [libmc.h](#).

### 13.38.2 Define Documentation

#### 13.38.2.1 `#define EXPORTMC`

Definition at line 147 of file libmc.h.

#### 13.38.2.2 `#define MC_Wait(arg1) MC_MainLoop(arg1)`

Definition at line 152 of file libmc.h.

#### 13.38.2.3 `#define THREAD_T pthread_t`

Definition at line 133 of file libmc.h.

Referenced by `acc_Thread()`, and `message_Send()`.

### 13.38.3 Typedef Documentation

#### 13.38.3.1 `typedef agency_t* agency_p`

Definition at line 242 of file libmc.h.

#### 13.38.3.2 `typedef struct agency_s agency_t`

The agency handle.

#### 13.38.3.3 `typedef agent_t* agent_p`

Definition at line 275 of file libmc.h.

#### 13.38.3.4 `typedef struct agent_s agent_t`

Definition at line 273 of file libmc.h.

#### 13.38.3.5 `typedef struct agent_thread_arg_s agent_thread_arg_t`

#### 13.38.3.6 `typedef enum error_code_e error_code_t`

#### 13.38.3.7 `typedef enum MC_SteerCommand_e MC_SteerCommand_t`

Available commands for MC\_Steer.

### 13.38.3.8 typedef agency\_p MCAgency\_t

Definition at line 243 of file libmc.h.

### 13.38.3.9 typedef struct MCAgencyOptions\_s MCAgencyOptions\_t

User modifiable agency [options](#).

### 13.38.3.10 typedef agent\_t\* MCAgent\_t

Definition at line 274 of file libmc.h.

## 13.38.4 Enumeration Type Documentation

### 13.38.4.1 enum error\_code\_e

Enumerator:

*MC\_SUCCESS*  
*MC\_ERR*  
*MC\_ERR\_CONNECT*  
*MC\_ERR\_PARSE*  
*MC\_ERR\_EMPTY*  
*MC\_ERR\_INVALID*  
*MC\_ERR\_INVALID\_ARGS*  
*MC\_ERR\_NOT\_FOUND*  
*MC\_ERR\_MEMORY*  
*MC\_ERR\_SEND*  
*MC\_WARN\_DUPLICATE*  
*MC\_SUCCESS*  
*ERR*  
*MC\_ERR\_CONNECT*  
*MC\_ERR\_PARSE*  
*MC\_ERR\_EMPTY*  
*MC\_ERR\_INVALID*  
*MC\_ERR\_INVALID\_ARGS*  
*MC\_ERR\_NOT\_FOUND*  
*MC\_ERR\_MEMORY*  
*MC\_ERR\_SEND*  
*MC\_WARN\_DUPLICATE*

Definition at line 166 of file libmc.h.

### 13.38.4.2 enum fipa\_performative\_e

Enumerator:

*FIPA\_ERROR*  
*FIPA\_ZERO*  
*FIPA\_ACCEPT\_PROPOSAL*  
*FIPA\_AGREE*  
*FIPA\_CANCEL*  
*FIPA\_CALL\_FOR\_PROPOSAL*  
*FIPA\_CONFIRM*  
*FIPA\_DISCONFIRM*  
*FIPA\_FAILURE*  
*FIPA\_INFORM*  
*FIPA\_INFORM\_IF*  
*FIPA\_INFORM\_REF*  
*FIPA\_NOT\_UNDERSTOOD*  
*FIPA\_PROPOGATE*  
*FIPA\_PROPOSE*  
*FIPA\_PROXY*  
*FIPA\_QUERY\_IF*  
*FIPA\_QUERY\_REF*  
*FIPA\_REFUSE*  
*FIPA\_REJECT\_PROPOSAL*  
*FIPA\_REQUEST*  
*FIPA\_REQUEST\_WHEN*  
*FIPA\_REQUEST\_WHENEVER*  
*FIPA\_SUBSCRIBE*  
*FIPA\_ERROR*  
*FIPA\_ZERO*  
*FIPA\_ACCEPT\_PROPOSAL*  
*FIPA\_AGREE*  
*FIPA\_CANCEL*  
*FIPA\_CALL\_FOR\_PROPOSAL*  
*FIPA\_CONFIRM*  
*FIPA\_DISCONFIRM*  
*FIPA\_FAILURE*  
*FIPA\_INFORM*  
*FIPA\_INFORM\_IF*  
*FIPA\_INFORM\_REF*  
*FIPA\_NOT\_UNDERSTOOD*  
*FIPA\_PROPOGATE*  
*FIPA\_PROPOSE*



*FIPA\_PROXY*  
*FIPA\_QUERY\_IF*  
*FIPA\_QUERY\_REF*  
*FIPA\_REFUSE*  
*FIPA\_REJECT\_PROPOSAL*  
*FIPA\_REQUEST*  
*FIPA\_REQUEST\_WHEN*  
*FIPA\_REQUEST\_WHENEVER*  
*FIPA\_SUBSCRIBE*

Definition at line 297 of file libmc.h.

#### 13.38.4.3 enum MC\_AgentStatus\_e

An agent's current execution state.

##### Enumerator:

*MC\_NO\_STATUS*  
*MC\_WAIT\_CH* Waiting to be started  
*MC\_WAIT\_MESSGSEND* Finished, waiting to migrate  
*MC\_AGENT\_ACTIVE* Running  
*MC\_AGENT\_NEUTRAL* Not running, but do not flush  
*MC\_AGENT\_SUSPENDED* Unused  
*MC\_WAIT\_FINISHED* Finished, waiting to be flushed

Definition at line 283 of file libmc.h.

#### 13.38.4.4 enum MC\_AgentType\_e

##### Enumerator:

*MC\_NONE*  
*MC\_REMOTE\_AGENT*  
*MC\_LOCAL\_AGENT*  
*MC\_RETURN\_AGENT*

Definition at line 278 of file libmc.h.

#### 13.38.4.5 enum MC\_Signal\_e

MobileC system signals.

##### Note:

Each signal is activated after the corresponding action. i.e. The 'MC\_RECV\_MESSAGE' signal is activated after a message is received.

See also:

[MC\\_WaitSignal\(\)](#), [MC\\_ResetSignal](#)

**Enumerator:**

*MC\_NO\_SIGNAL* No signal has been received  
*MC\_RECV\_CONNECTION* A Connection has been received  
*MC\_RECV\_MESSAGE* A Message has been received from a connection  
*MC\_RECV\_AGENT* An normal agent has been parsed from a message  
*MC\_RECV\_RETURN* A return agent has been parsed from a message  
*MC\_EXEC\_AGENT* A normal agent has been executed.  
*MC\_ALL\_SIGNALS* Catch any of the above signals

Definition at line 211 of file libmc.h.

#### 13.38.4.6 enum MC\_SteerCommand\_e

Available commands for MC\_Steer.

**Enumerator:**

*MC\_RUN* Continue the algorithm  
*MC\_SUSPEND* Suspend/pause the algorithm  
*MC\_RESTART* Restart the algorithm from the beginning  
*MC\_STOP* Stop the algorithm

Definition at line 196 of file libmc.h.

#### 13.38.4.7 enum MC\_ThreadIndex\_e

MobileC Module indices.

**Enumerator:**

*MC\_THREAD\_DF* Directory Facilitator  
*MC\_THREAD\_AMS* Agent Managment system  
*MC\_THREAD\_ACC* Agency communications  
*MC\_THREAD\_CP* Command Prompt  
*MC\_THREAD\_AGENT* Agent threads  
*MC\_THREAD\_ALL*

Definition at line 184 of file libmc.h.

### 13.38.5 Function Documentation

#### 13.38.5.1 EXPORTMC int MC\_AclAddReceiver (struct fipa\_acl\_message\_s \* *acl*, const char \* *name*, const char \* *address*)

Definition at line 279 of file libmc.c.

References `fipa_agent_identifier_s::addresses`, `fipa_agent_identifier_New()`, `fipa_agent_identifier_set_New()`, `fipa_agent_identifier_set_s::fipa_agent_identifiers`, `fipa_url_New()`, `fipa_url_sequence_New()`, `fipa_agent_identifier_s::name`, `fipa_url_sequence_s::num`, `fipa_agent_identifier_set_s::num`, `fipa_acl_message_s::receiver`, `fipa_acl_message_s::receiver_num`, `fipa_url_s::str`, and `fipa_url_sequence_s::urls`.

Referenced by `MC_AclAddReceiver_chdl()`.

### 13.38.5.2 EXPORTMC int MC\_AclAddReplyTo (struct fipa\_acl\_message\_s \* *acl*, const char \* *name*, const char \* *address*)

Definition at line 315 of file `libmc.c`.

References `fipa_agent_identifier_s::addresses`, `fipa_agent_identifier_New()`, `fipa_agent_identifier_set_New()`, `fipa_agent_identifier_set_s::fipa_agent_identifiers`, `fipa_url_New()`, `fipa_url_sequence_New()`, `fipa_agent_identifier_s::name`, `fipa_url_sequence_s::num`, `fipa_agent_identifier_set_s::num`, `fipa_acl_message_s::reply_to`, `fipa_url_s::str`, and `fipa_url_sequence_s::urls`.

Referenced by `MC_AclAddReplyTo_chdl()`.

### 13.38.5.3 EXPORTMC int MC\_AclDestroy (struct fipa\_acl\_message\_s \* *message*)

Destroy a FIPA ACL message.

#### Parameters:

*message* The ACL message to destroy

#### Returns:

0 on success, error code on failure.

Definition at line 78 of file `libmc.c`.

References `fipa_acl_message_Destroy()`.

Referenced by `MC_AclDestroy_chdl()`.

### 13.38.5.4 EXPORTMC struct fipa\_acl\_message\_s\* MC\_AclNew (void) [read]

Allocate a new ACL Message.

#### Returns:

A newly allocated and empty ACL message.

Definition at line 84 of file `libmc.c`.

References `fipa_acl_message_New()`.

Referenced by `MC_AclNew_chdl()`.

### 13.38.5.5 EXPORTMC int MC\_AclPost (MCAgent\_t *agent*, struct fipa\_acl\_message\_s \* *message*)

Post ACL message to agent.

**Parameters:**

*agent* The agent to post the message to

*message* The message to post

**Returns:**

0 if successful, or error\_code\_t type.

Definition at line 89 of file libmc.c.

References agent\_mailbox\_Post(), and agent\_s::mailbox.

Referenced by MC\_AclPost\_chdl(), and MC\_AclSend().

### 13.38.5.6 EXPORTMC struct fipa\_acl\_message\_s\* MC\_AclReply (struct fipa\_acl\_message\_s \* *acl\_message*) [read]

Reply to an ACL message.

**Parameters:**

*acl\_message* The incoming acl message to reply to

**Returns:**

A newly allocated ACL message

**Note:**

This function simply generates a new ACL message with the 'receiver' field automatically set to the 'sender' field of the incoming message.

Definition at line 95 of file libmc.c.

References fipa\_Reply().

Referenced by MC\_AclReply\_chdl().

### 13.38.5.7 EXPORTMC struct fipa\_acl\_message\_s\* MC\_AclRetrieve (MCAgent\_t *agent*) [read]

Retrieve an ACL message.

**Parameters:**

*agent* Agent to retrieve message from.

**Returns:**

an ACL message struct on success or NULL on failure

Definition at line 101 of file libmc.c.

References agent\_mailbox\_Retrieve(), and agent\_s::mailbox.

Referenced by MC\_AclRetrieve\_chdl().

**13.38.5.8 EXPORTMC int MC\_AclSend (MC\_Agency\_t *attr*, struct fipa\_acl\_message\_s \* *acl*)**

Send a composed ACL Message.

**Parameters:**

*attr* An initialized and running MobileC agency

*acl* An allocated and fully composed ACL message.

**Returns:**

0 if successful, error code on failure.

Definition at line 107 of file libmc.c.

References fipa\_agent\_identifier\_s::addresses, mtp\_http\_s::content, mtp\_http\_content\_s::content\_type, mtp\_http\_content\_s::data, dynstring\_Destroy(), FIPA\_ACL, fipa\_acl\_Compose(), fipa\_acl\_message\_Copy(), fipa\_agent\_identifier\_set\_s::fipa\_agent\_identifiers, fipa\_envelope\_Compose(), mtp\_http\_s::host, http\_to\_hostport(), MC\_AclPost(), MC\_FindAgentByName(), agency\_s::mc\_platform, dynstring\_s::message, mtp\_http\_s::message\_parts, message\_Send(), message\_s::message\_type, mtp\_http\_CreateMessage(), mtp\_http\_Destroy(), mtp\_http\_New(), fipa\_agent\_identifier\_s::name, fipa\_url\_sequence\_s::num, fipa\_agent\_identifier\_set\_s::num, port, mc\_platform\_s::private\_key, fipa\_acl\_message\_s::receiver, fipa\_url\_s::str, mtp\_http\_s::target, message\_s::target, and fipa\_url\_sequence\_s::urls.

Referenced by MC\_AclSend\_chdl().

**13.38.5.9 EXPORTMC int MC\_AclSetContent (struct fipa\_acl\_message\_s \* *acl*, const char \* *content*)**

Definition at line 350 of file libmc.c.

References fipa\_string\_s::content, fipa\_acl\_message\_s::content, fipa\_string\_Destroy(), and fipa\_string\_New().

Referenced by MC\_AclSetContent\_chdl().

**13.38.5.10 EXPORTMC int MC\_AclSetConversationID (struct fipa\_acl\_message\_s \* *acl*, char \* *id*)**

Definition at line 229 of file libmc.c.

References fipa\_string\_s::content, fipa\_expression\_s::content, fipa\_acl\_message\_s::conversation\_id, FIPA\_EXPR\_STRING, fipa\_expression\_New(), fipa\_string\_New(), fipa\_expression\_s::content\_u::string, and fipa\_expression\_s::type.

Referenced by MC\_AclSetConversationID\_chdl().

**13.38.5.11 EXPORTMC int MC\_AclSetPerformative (struct fipa\_acl\_message\_s \* *acl*, enum fipa\_performative\_e *performative*)**

Definition at line 248 of file libmc.c.

References fipa\_acl\_message\_s::performative.

Referenced by MC\_AclSetPerformative\_chdl().

### 13.38.5.12 EXPORTMC int MC\_AclSetProtocol (struct fipa\_acl\_message\_s \* *acl*, enum fipa\_protocol\_e *performative*)

Definition at line 213 of file libmc.c.

References FIPA\_PROTOCOL\_END, FIPA\_PROTOCOL\_ERROR, and fipa\_acl\_message\_s::protocol.

Referenced by MC\_AclSetProtocol\_chdl().

### 13.38.5.13 EXPORTMC int MC\_AclSetSender (struct fipa\_acl\_message\_s \* *acl*, const char \* *name*, const char \* *address*)

Definition at line 256 of file libmc.c.

References fipa\_agent\_identifier\_s::addresses, fipa\_agent\_identifier\_Destroy(), fipa\_agent\_identifier\_New(), fipa\_url\_New(), fipa\_url\_sequence\_New(), fipa\_agent\_identifier\_s::name, fipa\_url\_sequence\_s::num, fipa\_acl\_message\_s::sender, fipa\_url\_s::str, and fipa\_url\_sequence\_s::urls.

Referenced by MC\_AclSetSender\_chdl().

### 13.38.5.14 EXPORTMC struct fipa\_acl\_message\_s\* MC\_AclWaitRetrieve (MCAgent\_t *agent*) [read]

Wait for and retrieve an ACL message.

#### Parameters:

*agent* Agent to retrieve message from.

#### Returns:

an ACL message struct on success or NULL on failure

Definition at line 206 of file libmc.c.

References agent\_mailbox\_WaitRetrieve(), and agent\_s::mailbox.

Referenced by MC\_AclWaitRetrieve\_chdl().

### 13.38.5.15 EXPORTMC int MC\_AddAgent (MCAgency\_t *attr*, MCAgent\_t *agent*)

Add an agent to the agency 'attr'.

#### Parameters:

*attr* a MobileC agency

*agent* An initialized MobileC agent

#### Returns:

0 if successful, or error\_code\_t type

Definition at line 367 of file libmc.c.

References mc\_platform\_s::agent\_queue, mc\_platform\_s::ams, COND\_SIGNAL, agency\_s::mc\_platform, agent\_s::mc\_platform, MUTEX\_LOCK, and MUTEX\_UNLOCK.

Referenced by MC\_AddAgent\_chdl().

**13.38.5.16** `int MC_AddStationaryAgent (MCAgency_t agency, void (*)(struct agent_thread_arg_s *) agent_thread, const char * name, void * agent_args)`

Definition at line 381 of file libmc.c.

References agent\_thread\_arg\_s::agent, agent\_NewBinary(), mc\_platform\_s::agent\_queue, agent\_thread\_arg\_s::args, agent\_thread\_arg\_s::attr, agency\_s::mc\_platform, agent\_s::name, agent\_thread\_arg\_s::thread, and THREAD\_CREATE.

**13.38.5.17** `const void* MC_AgentVariableRetrieve (MCAgent_t agent, const char * var_name, int task_num)`

Retrieve a pointer to a previously saved variable.

#### Parameters:

*agent* A MobileC agent.

*var\_name* The name of the saved variable that has previously been saved.

*task\_num* The previous completed task from which to retrieve the saved variable.

#### Returns:

A pointer to the data on success or NULL on failure.

## 13.38.6 Examples

The following example demonstrates usage of [MC\\_AgentVariableRetrieve\(\)](#) from agent space.

Definition at line 402 of file libmc.c.

References agent\_task\_s::agent\_variable\_list, interpreter\_variable\_data\_s::data, agent\_s::datastate, agent\_datastate\_s::task\_progress, and agent\_datastate\_s::tasks.

Referenced by MC\_AgentVariableRetrieve\_chdl().

**13.38.6.1** `int MC_AgentVariableRetrieveInfo (MCAgent_t agent, const char * var_name, int task_num, const void ** data, int * dim, const int ** extent)`

Retrieve a info about a previously saved variable.

#### Parameters:

*agent* A MobileC agent.

*var\_name* The name of the saved variable that has previously been saved.

*task\_num* The previous completed task from which to retrieve the saved variable.

*data* (Output) The Variable Data

*dim* (Output) The dimension of the data array

*extent* (Output) The extents of the output array

#### Returns:

Error code.

Definition at line 420 of file libmc.c.

References `agent_task_s::agent_variable_list`, `interpreter_variable_data_s::array_dim`, `interpreter_variable_data_s::array_extent`, `interpreter_variable_data_s::data`, `agent_s::datastate`, `MC_ERR_NOT_FOUND`, `MC_SUCCESS`, `agent_datastate_s::task_progress`, and `agent_datastate_s::tasks`.

### 13.38.6.2 `int MC_AgentVariableSave (MCAgent_t agent, const char * var_name)`

Mark an agent variable for saving.

#### Parameters:

*agent* A MobileC agent.

*var\_name* The name of the variable to mark for saving.

#### Returns:

0 on success, non-zero on failure.

#### See also:

`test1.xml`

## 13.38.7 Examples

See `agent_saved_variables_example/test1.xml` for an example of usage of this api function.

## 13.38.8 Examples

The following example demonstrates usage of `MC_AgentVariableSave()` from agent space.

Definition at line 441 of file libmc.c.

References `agent_s::datastate`, `MC_ERR_MEMORY`, `agent_task_s::num_saved_variables`, `agent_task_s::saved_variables`, `agent_datastate_s::task_progress`, and `agent_datastate_s::tasks`.

Referenced by `MC_AgentVariableSave_chdl()`.

### 13.38.8.1 `EXPORTMC int MC_BarrierDelete (MCAgency_t attr, int id)`

Find and delete an initialized MobileC Barrier.

#### Parameters:

*attr* A running MobileC agency

*id* The id of the barrier node to delete

#### Returns:

returns 0 on success, error if the node is not found or other failure.

Definition at line 502 of file libmc.c.

References `mc_platform_s::barrier_queue`, `barrier_queue_Delete()`, and `agency_s::mc_platform`.

Referenced by `MC_BarrierDelete_chdl()`.



### 13.38.8.2 EXPORTMC int MC\_BarrierInit (MCAgency\_t *attr*, int *id*, int *num\_procs*)

Initialize a MobileC Barrier.

#### Parameters:

*attr* A running MobileC agency  
*id* The requested barrier id  
*num\_procs* The number of agents/threads/processes that will wait on the barrier

#### Returns:

The allocated barrier id. May differ from the requested id if it is already in use.

## 13.38.9 Examples

The following example demonstrates an agent which sets up an MC\_Barrier.

Definition at line 488 of file libmc.c.

References `barrier_node_Initialize()`, `mc_platform_s::barrier_queue`, `barrier_queue_Add()`, `barrier_queue_Get()`, `MC_ERR`, `agency_s::mc_platform`, `MC_SUCCESS`, and `node`.

Referenced by `MC_BarrierInit_chdl()`.

### 13.38.9.1 EXPORTMC int MC\_CallAgentFunc (MCAgent\_t *agent*, const char \* *funcName*, void \* *returnVal*, int *numArgs*, ...)

Use custom ChOptions\_t type for internal Ch interpreter.

#### Parameters:

*attr* A running MobileC agency  
*options* Initialized Ch *options* structure

#### Returns:

0 on success, error\_code\_t type on failure Calls a function defined in an agent

#### Parameters:

*agent* An initialized and executed MobileC agent  
*funcName* The name of the function to call  
*returnVal* The agent function's return value  
*numArgs* The number of arguments supplied to the agent function  
... A variable argument list to be supplied to the agent function

#### Returns:

0 if successful, error\_code\_t type on failure

### 13.38.10 Example

Definition at line 508 of file libmc.c.

References agent\_s::agent\_interp, MUTEX\_LOCK, MUTEX\_UNLOCK, and agent\_s::run\_lock.

**13.38.10.1 EXPORTMC int MC\_CallAgentFuncArg (MCAgent\_t *agent*, const char \**funcName*, void \**returnVal*, void \**arg*)**

Calls a function defined in an agent.

**Parameters:**

*agent* An initialized and executed MobileC agent  
*funcName* The name of the function to call  
*returnVal* The agent function's return value  
*arg* The agent functions argument

**Note:**

The agent function must be of the form 'void\* func(void\* arg);'

**Returns:**

0 if successful, error\_code\_t type on failure

### 13.38.11 Example

Definition at line 529 of file libmc.c.

References agent\_s::agent\_interp, MUTEX\_LOCK, MUTEX\_UNLOCK, and agent\_s::run\_lock.

**13.38.11.1 EXPORTMC int MC\_CallAgentFuncV (MCAgent\_t *agent*, const char \**funcName*, void \**returnVal*, va\_list *ap*)**

Calls a function defined in an agent.

**Parameters:**

*agent* An initialized and executed MobileC agent  
*funcName* The name of the function to call  
*returnVal* The agent function's return value  
*ap* A variable argument list

**Returns:**

0 if successful, error\_code\_t type on failure

Definition at line 548 of file libmc.c.

References agent\_s::agent\_interp, MUTEX\_LOCK, MUTEX\_UNLOCK, and agent\_s::run\_lock.

**13.38.11.2 EXPORTMC int MC\_CallAgentFuncVar (MCAgent\_t *agent*, const char \* *funcName*, void \* *returnVal*, ChVaList\_t *arglist*)**

Definition at line 568 of file libmc.c.

References agent\_s::agent\_interp, MUTEX\_LOCK, MUTEX\_UNLOCK, and agent\_s::run\_lock.

Referenced by MC\_CallAgentFunc\_chdl().

**13.38.11.3 EXPORTMC MCAgent\_t MC\_ComposeAgent (const char \* *name*, const char \* *home*, const char \* *owner*, const char \* *code*, const char \* *return\_var\_name*, const char \* *server*, int *persistent*)**

Compose a new agent dynamically without using a prewritten XML file.

**Parameters:**

*name* The desired name of the new agent.

*home* The home of the new agent.

*owner* The owner of the new agent.

*code* The agent code

*return\_var\_name* The name of the agent's return variable. Set to "no-return" if no return variable is desired.

*server* The target destination of the agent.

*persitent* A flag indicating whether or not the agent should be persistent. A value of '1' indicates persistence, while a value of '0' indicates default non-persistent behaviour.

**Returns:**

This function returns a valid MCAgent\_t structure on success or NULL on failure.

Definition at line 608 of file libmc.c.

References MC\_ComposeAgentS().

**13.38.11.4 EXPORTMC MCAgent\_t MC\_ComposeAgentFromFile (const char \* *name*, const char \* *home*, const char \* *owner*, const char \* *filename*, const char \* *return\_var\_name*, const char \* *server*, int *persistent*)**

Compose a new agent dynamically from a source code file.

**Parameters:**

*filename* The filename of the source file

*name* The desired name of the new agent.

*home* The home of the new agent.

*owner* The owner of the new agent.

*code* The agent code

*return\_var\_name* The name of the agent's return variable. Set to "no-return" if no return variable is desired.

*server* The target destination of the agent.

*persistent* A flag indicating whether or not the agent should be persistent. A value of '1' indicates persistence, while a value of '0' indicates default non-persistent behaviour.

**Returns:**

This function returns a valid MCAgent\_t structure on success or NULL on failure.

Definition at line 707 of file libmc.c.

References MC\_ComposeAgentFromFileS().

### 13.38.11.5 EXPORTMC MCAgent\_t MC\_ComposeAgentFromFileS (const char \* *name*, const char \* *home*, const char \* *owner*, const char \* *filename*, const char \* *return\_var\_name*, const char \* *server*, const char \* *workgroup\_code*, int *persistent*)

Compose a new agent dynamically from a source code file.

**Parameters:**

*filename* The filename of the source file

*name* The desired name of the new agent.

*home* The home of the new agent.

*owner* The owner of the new agent.

*code* The agent code

*return\_var\_name* The name of the agent's return variable. Set to "no-return" if no return variable is desired.

*server* The target destination of the agent.

*workgroup\_code* The secret workgroup code of the agent. Only agents with the same workgroup code may perform certain interactions.

*persistent* A flag indicating whether or not the agent should be persistent. A value of '1' indicates persistence, while a value of '0' indicates default non-persistent behaviour.

**Returns:**

This function returns a valid MCAgent\_t structure on success or NULL on failure.

Definition at line 728 of file libmc.c.

References agent\_thread\_arg\_s::agent, and MC\_ComposeAgentS().

Referenced by MC\_ComposeAgentFromFile().

### 13.38.11.6 EXPORTMC MCAgent\_t MC\_ComposeAgentS (const char \* *name*, const char \* *home*, const char \* *owner*, const char \* *code*, const char \* *return\_var\_name*, const char \* *server*, const char \* *workgroup\_code*, int *persistent*)

Compose a new agent dynamically without using a prewritten XML file.

**Parameters:**

*name* The desired name of the new agent.

*home* The home of the new agent.

**owner** The owner of the new agent.

**code** The agent code

**return\_var\_name** The name of the agent's return variable. Set to "no-return" if no return variable is desired.

**server** The target destination of the agent.

**workgroup\_code** The secret workgroup code of the agent. Only agents with the same workgroup code may perform certain interactions.

**persistent** A flag indicating whether or not the agent should be persistent. A value of '1' indicates persistence, while a value of '0' indicates default non-persistent behaviour.

#### Returns:

This function returns a valid MCAgent\_t structure on success or NULL on failure.

Definition at line 630 of file libmc.c.

References agent\_thread\_arg\_s::agent, agent\_datastate\_s::agent\_code, agent\_datastate\_s::agent\_code\_ids, agent\_datastate\_s::agent\_codes, agent\_datastate\_New(), agent\_New(), agent\_s::agent\_status, agent\_task\_New(), agent\_s::agent\_type, agent\_s::datastate, agent\_s::home, MC\_LOCAL\_AGENT, MC\_WAIT\_MESSGSEND, agent\_s::name, agent\_datastate\_s::number\_of\_tasks, agent\_s::orphan, agent\_s::owner, agent\_datastate\_s::persistent, agent\_task\_s::server\_name, agent\_datastate\_s::tasks, agent\_task\_s::var\_name, and agent\_s::wg\_code.

Referenced by MC\_ComposeAgent(), MC\_ComposeAgent\_chdl(), MC\_ComposeAgentFromFileS(), and MC\_ComposeAgentS\_chdl().

#### 13.38.11.7 EXPORTMC int MC\_CondBroadcast (MCAgency\_t attr, int id)

Wakes up all agents/threads waiting on a condition variable.

#### Parameters:

**attr** A MobileC agency

**id** Synchronization variable id to broadcast to

#### See also:

[MC\\_SyncInit\(\)](#), [MC\\_CondSignal\(\)](#)

#### Returns:

0 on success, error\_code\_t type on failure

Definition at line 767 of file libmc.c.

References syncListNode\_s::cond, COND\_BROADCAST, syncListNode\_s::lock, MC\_ERR\_NOT\_FOUND, agency\_s::mc\_platform, MUTEX\_LOCK, MUTEX\_UNLOCK, syncListNode\_s::signalled, mc\_platform\_s::syncList, and syncListFind().

Referenced by MC\_CondBroadcast\_chdl().

#### 13.38.11.8 EXPORTMC int MC\_CondReset (MCAgency\_t attr, int id)

Reset a previously signalled MobileC condition variable.

**Parameters:**

*attr* A MobileC Agency

*id* The synchronization variable id to reset

**See also:**

[MC\\_SyncInit\(\)](#)

**Returns:**

0 on success, error\_code\_t type on failure

Definition at line 819 of file libmc.c.

References syncListNode\_s::lock, MC\_ERR\_NOT\_FOUND, agency\_s::mc\_platform, MUTEX\_LOCK, MUTEX\_UNLOCK, syncListNode\_s::signalled, mc\_platform\_s::syncList, and syncListFind().

Referenced by MC\_CondReset\_chdl().

### 13.38.11.9 EXPORTMC int MC\_CondSignal (MCAgency\_t *attr*, int *id*)

Wakes up at least one thread waiting on a condition variable.

**Parameters:**

*attr* A MobileC agency

*id* synchronization variable id

**See also:**

[MC\\_SyncInit\(\)](#)

**Returns:**

0 on success, error\_code\_t type on failure

### 13.38.12 Example

The following example demonstrates the agent-space version of the function, which is nearly identical to the binary space api function.

Definition at line 782 of file libmc.c.

References syncListNode\_s::cond, COND\_SIGNAL, syncListNode\_s::lock, MC\_ERR\_NOT\_FOUND, agency\_s::mc\_platform, MUTEX\_LOCK, MUTEX\_UNLOCK, syncListNode\_s::signalled, mc\_platform\_s::syncList, and syncListFind().

Referenced by MC\_CondSignal\_chdl().

#### 13.38.12.1 EXPORTMC int MC\_CondWait (MCAgency\_t *attr*, int *id*)

Wait on a MobileC synchronization variable.

**Parameters:**

*attr* A MobileC agency  
*id* a synchronization variable id

**See also:**

[MC\\_SyncInit\(\)](#)

**Returns:**

0 on success, error\_code\_t type on failure

### 13.38.13 Example

The following example demonstrates the agent-space version of this function.

Definition at line 797 of file libmc.c.

References syncListNode\_s::cond, COND\_WAIT, syncListNode\_s::lock, MC\_ERR\_NOT\_FOUND, agency\_s::mc\_platform, MUTEX\_LOCK, MUTEX\_UNLOCK, syncListNode\_s::signalled, mc\_platform\_s::syncList, and syncListFind().

Referenced by MC\_CondWait\_chdl().

#### 13.38.13.1 int MC\_CopyAgent (MCAgent\_t \* *agent\_out*, const MCAgent\_t *agent\_in*)

Performs a deep-copy of an agent structure.

**Parameters:**

*agent\_out* A pointer to the agent to copy to.  
*agent\_in* The agent to copy

**Returns:**

0 on success, error\_code\_t type on failure.

Definition at line 837 of file libmc.c.

References agent\_Copy(), and MC\_SUCCESS.

#### 13.38.13.2 EXPORTMC int MC\_DeleteAgent (MCAgent\_t *agent*)

Stop and remove an agent.

**Parameters:**

*agent* An agent in any state (running, waiting, etc)

**Returns:**

0 on success, error\_code\_t type on failure

Definition at line 844 of file libmc.c.

References CHECK\_NULL, MC\_ERR\_INVALID, MC\_ERR\_INVALID\_ARGS, MC\_SetAgentStatus(), MC\_SUCCESS, MC\_TerminateAgent(), MC\_WAIT\_FINISHED, and agent\_s::wg\_code.

Referenced by MC\_DeleteAgent\_chdl().

### 13.38.13.3 EXPORTMC int MC\_DeleteAgentWG (MCAgent\_t *calling\_agent*, MCAgent\_t *agent*)

Stop and remove an agent in the same workgroup.

#### Parameters:

*calling\_agent* The calling agent  
*agent* An agent in any state (running, waiting, etc)

#### Note:

The agents must belong to the same workgroup.

#### Returns:

0 on success, error\_code\_t type on failure

Definition at line 864 of file libmc.c.

References CHECK\_NULL, MC\_ERR\_INVALID, MC\_ERR\_INVALID\_ARGS, MC\_SetAgentStatus(), MC\_SUCCESS, MC\_TerminateAgentWG(), MC\_WAIT\_FINISHED, and agent\_s::wg\_code.

Referenced by MC\_DeleteAgentWG\_chdl().

### 13.38.13.4 int MC\_DestroyServiceSearchResult (char \*\* *agentName*, char \*\* *serviceName*, int \* *agentID*, int *numResult*)

Free memory allocated by a Service Search operation.

#### Parameters:

*agentName* agent names returned by a search operation.  
*serviceName* service names return by a search operation.  
*agentID* list of agent id's returned by a search operation.  
*numResult* The number of hits returned by a search operation.

#### Returns:

0 on success, error code on failure.

Definition at line 887 of file libmc.c.

Referenced by MC\_DestroyServiceSearchResult\_chdl().

### 13.38.13.5 EXPORTMC int MC\_End (MCAgency\_t *attr*)

End an agency.



**Parameters:**

*attr* A running agency

**Returns:**

0 on success, `error_code_t` type on failure

**13.38.14 Example**

Definition at line 936 of file `libmc.c`.

References `mc_platform_s::acc`, `mc_platform_s::ams`, `mc_platform_s::cmd_prompt`, `COND_BROADCAST`, `COND_SIGNAL`, `mc_platform_s::connection_queue`, `mc_platform_s::df`, `GET_THREAD_MODE`, `agency_s::hostName`, `agency_s::mc_platform`, `mc_platform_Destroy()`, `MC_THREAD_ACC`, `MC_THREAD_AMS`, `MC_THREAD_CP`, `MC_THREAD_DF`, `mc_platform_s::message_queue`, `MUTEX_LOCK`, `MUTEX_UNLOCK`, `mc_platform_s::quit`, `mc_platform_s::quit_cond`, `mc_platform_s::quit_lock`, `cmd_prompt_s::thread`, `THREAD_CANCEL`, `THREAD_JOIN`, and `agency_s::threads`.

Referenced by `MC_End_chdl()`.

**13.38.14.1 EXPORTMC MCAgent\_t MC\_FindAgentByID (MCAgency\_t *attr*, int *ID*)**

Find an agent by its id.

**Parameters:**

*attr* the agency to search

*ID* the id to search for

**Returns:**

a valid agent on success, NULL on failure

Definition at line 1012 of file `libmc.c`.

References `mc_platform_s::agent_queue`, and `agency_s::mc_platform`.

Referenced by `MC_FindAgentByID_chdl()`.

**13.38.14.2 EXPORTMC MCAgent\_t MC\_FindAgentByName (MCAgency\_t *attr*, const char \* *name*)**

Find an agent by its name.

**Parameters:**

*attr* a running agency

*name* name to search for

**Returns:**

a valid agent on success or NULL on failure

### 13.38.15 Example

Definition at line 999 of file libmc.c.

References mc\_platform\_s::agent\_queue, and agency\_s::mc\_platform.

Referenced by MC\_AclSend(), MC\_DeleteAgent\_chdl(), MC\_DeleteAgentWG\_chdl(), MC\_FindAgentByName\_chdl(), MC\_TerminateAgent\_chdl(), and MC\_TerminateAgentWG\_chdl().

#### 13.38.15.1 EXPORTMC void\* MC\_GetAgentExecEngine (MCAgent\_t agent)

Retrieve an agent's Ch interpreter.

**Parameters:**

*agent* a valid agent

**Returns:**

a Ch interpreter of type 'ChInterp\_t' on success, or NULL on failure.

Definition at line 1066 of file libmc.c.

References agent\_s::agent\_interp.

#### 13.38.15.2 EXPORTMC int MC\_GetAgentID (MCAgent\_t agent)

Retrieve an agent's id.

Definition at line 1072 of file libmc.c.

References agent\_s::id.

Referenced by MC\_GetAgentID\_chdl().

#### 13.38.15.3 EXPORTMC char\* MC\_GetAgentName (MCAgent\_t agent)

Definition at line 1080 of file libmc.c.

References agent\_s::lock, MUTEX\_LOCK, MUTEX\_UNLOCK, and agent\_s::name.

Referenced by MC\_GetAgentName\_chdl().

#### 13.38.15.4 EXPORTMC int MC\_GetAgentNumTasks (MCAgent\_t agent)

Retrieve the number of tasks an agent has.

### 13.38.16 Example

Definition at line 1161 of file libmc.c.

References agent\_s::datastate, and agent\_datastate\_s::number\_of\_tasks.

Referenced by MC\_GetAgentNumTasks\_chdl().

**13.38.16.1 EXPORTMC int MC\_GetAgentReturnData (MCAgent\_t agent, int task\_num, void \*\* data, int \* dim, int \*\* extent)**

Get an agent's return data.

**Parameters:**

*agent* a valid agent

*task\_num* the task for which to retrieve the return data. The task must already be completed.

*data* the return data. May be multi dimensional array.

*dim* the number of dimensions of the return array.

*extent* the extent of each one of the array dimensions.

**13.38.17 Example**

This file demonstrates the retrieval of agent return data from an agent

This is the agent which gets the data

Definition at line 1098 of file libmc.c.

References agent\_task\_s::agent\_return\_data, interpreter\_variable\_data\_s::array\_dim, interpreter\_variable\_data\_s::array\_extent, CH\_DATATYPE\_SIZE, interpreter\_variable\_data\_s::data\_type, agent\_s::datastate, agent\_datastate\_s::number\_of\_tasks, size, and agent\_datastate\_s::tasks.

**13.38.17.1 EXPORTMC int MC\_GetAgentStatus (MCAgent\_t agent)**

Get an agent's current status.

**Returns:**

returns type 'enum MC\_AgentStatus\_e'

Definition at line 1044 of file libmc.c.

References agent\_s::agent\_status, agent\_s::lock, MUTEX\_LOCK, and MUTEX\_UNLOCK.

Referenced by MC\_GetAgentStatus\_chdl().

**13.38.17.2 EXPORTMC enum MC\_AgentType\_e MC\_GetAgentType (MCAgent\_t agent)**

Get an agent's type.

**Returns:**

returns type 'enum MC\_AgentType\_e'

Definition at line 1167 of file libmc.c.

References agent\_s::agent\_type.

**13.38.17.3 EXPORTMC char\* MC\_GetAgentXMLString (MCAgent\_t *agent*)**

Get an agent's xml string.

**Returns:**

a malloc'd character string containing the agent's xml code

Definition at line 1054 of file libmc.c.

References agent\_s::datastate, mxmlSaveAllocString(), and agent\_datastate\_s::xml\_agent\_root.

Referenced by MC\_GetAgentXMLString\_chdl().

**13.38.17.4 EXPORTMC int MC\_HaltAgency (MCAgency\_t *agency*)**

Halt an agency: Do not process new entries in queues.

**Parameters:**

*agency* A handle to a running MobileC agency.

**Returns:**

0 on success, non-zero on failure.

Definition at line 1213 of file libmc.c.

References mc\_platform\_s::giant, mc\_platform\_s::giant\_lock, agency\_s::mc\_platform, MUTEX\_LOCK, and MUTEX\_UNLOCK.

Referenced by MC\_GetAllAgents(), and MC\_HaltAgency\_chdl().

**13.38.17.5 EXPORTMC MCAgency\_t MC\_Initialize (int *port*, MCAgencyOptions\_t \* *options*)**

Initialize and start a MobileC agency.

**Parameters:**

*port* the TCP port the agency should bind to

*options* initialized MobileC [options](#) or NULL for default [options](#)

**Returns:**

a handle to a running MobileC agency or NULL on failure

**13.38.18 Example**

Definition at line 1222 of file libmc.c.

References mc\_platform\_s::agency, buf, MCAgencyOptions\_s::ch\_options, CHECK\_NULL, agency\_s::client, MCAgencyOptions\_s::default\_agent\_status, agency\_s::default\_agentstatus, f, HOST\_NAME\_MAX, agency\_s::hostName, MCAgencyOptions\_s::initInterps, agency\_s::initInterps, agency\_s::known\_host\_filename, MCAgencyOptions\_s::known\_host\_filename, MC\_InitializeAgencyOptions(), mc\_platform, agency\_s::mc\_platform, mc\_platform\_Initialize(), MC\_THREAD\_ALL, MCAgencyOptions\_s::passphrase, agency\_s::portno, agency\_s::priv\_key\_filename, MCAgencyOptions\_s::priv\_key\_filename,

read\_encrypted\_file(), agency\_s::server, MCAgencyOptions\_s::stack\_size, agency\_s::stack\_size, MCAgencyOptions\_s::threads, and agency\_s::threads.

### 13.38.18.1 EXPORTMC int MC\_InitializeAgencyOptions (struct MCAgencyOptions\_s \* *options*)

Initialize MobileC *options*.

#### Parameters:

*options* *options* to initialize.

#### Returns:

0 on success, error\_code\_t on failure

#### Note:

MobileC *options* should be initialized with this function before any of its members are modified.

## 13.38.19 Example

Definition at line 1333 of file libmc.c.

References MCAgencyOptions\_s::default\_agent\_status, MCAgencyOptions\_s::initInterps, MCAgencyOptions\_s::known\_host\_filename, MC\_THREAD\_ALL, MC\_WAIT\_CH, MCAgencyOptions\_s::modified, MCAgencyOptions\_s::passphrase, MCAgencyOptions\_s::priv\_key\_filename, MCAgencyOptions\_s::stack\_size, and MCAgencyOptions\_s::threads.

Referenced by MC\_Initialize().

### 13.38.19.1 EXPORTMC int MC\_LoadAgentFromFile (MCAgency\_t *attr*, const char \* *filename*)

Load an agent from a file into an agency.

#### Parameters:

*agency* A valid and running Mobile-C agency

*filename* Filename containing the agent to load

#### Returns:

0 on success, non-zero on failure.

Definition at line 1354 of file libmc.c.

References buf, agency\_s::mc\_platform, message\_Destroy(), message\_InitializeFromString(), message\_New(), mc\_platform\_s::message\_queue, MXML\_DESCEND, mxmlFindElement(), mxmlLoadString(), message\_s::to\_address, message\_s::xml\_payload, and message\_s::xml\_root.

### 13.38.19.2 EXPORTMC int MC\_MainLoop (MCAgency\_t *attr*)

Wait indefinitely.

**Note:**

This function is intended to block the calling thread forever.

Definition at line 2008 of file libmc.c.

References COND\_WAIT, agency\_s::mc\_platform, MUTEX\_LOCK, MUTEX\_UNLOCK, mc\_platform\_s::quit, mc\_platform\_s::quit\_cond, and mc\_platform\_s::quit\_lock.

**13.38.19.3 EXPORTMC int MC\_MigrateAgent (MCAgent\_t agent, const char \* hostname, int port)**

Migrates a running agent to another host.

**Parameters:**

*agent* The agent to migrate  
*hostname* The new host to migrate the agent to  
*port* The new port to migrate the agent to

**Returns:**

0 on success, error\_code\_t type on failure.

Definition at line 1424 of file libmc.c.

References agent\_s::datastate, agent\_datastate\_s::progress\_modifier, agent\_task\_s::server\_name, agent\_datastate\_s::task\_progress, and agent\_datastate\_s::tasks.

Referenced by MC\_MigrateAgent\_chdl().

**13.38.19.4 EXPORTMC int MC\_MutexLock (MCAgency\_t attr, int id)**

Locks a MobileC synchronization variable as a mutex.

**Parameters:**

*attr* a MobileC agency handle  
*id* the synchronization variable id to lock

**Returns:**

0 on success, error\_code\_t type on failure

**13.38.20 Example**

Consider the following agents, which use the agent-space version of this api function. Note that the 'sleep' agent is sent first, followed by the 'wake' agent.

Definition at line 1446 of file libmc.c.

References syncListNode\_s::lock, agency\_s::mc\_platform, MUTEX\_LOCK, mc\_platform\_s::syncList, and syncListFind().

Referenced by MC\_MutexLock\_chdl().

### 13.38.20.1 EXPORTMC int MC\_MutexUnlock (MCAgency\_t attr, int id)

Definition at line 1458 of file libmc.c.

References syncListNode\_s::lock, agency\_s::mc\_platform, MUTEX\_UNLOCK, mc\_platform\_s::syncList, and syncListFind().

Referenced by MC\_MutexUnlock\_chdl().

### 13.38.20.2 EXPORTMC int MC\_PrintAgentCode (MCAgent\_t agent)

Prints an agents code to stdout.

#### Returns:

0 on success, error\_code\_t on failure

Definition at line 1470 of file libmc.c.

References agent\_datastate\_s::agent\_code, agent\_s::datastate, agent\_s::lock, MUTEX\_LOCK, MUTEX\_UNLOCK, agent\_datastate\_s::number\_of\_tasks, and agent\_datastate\_s::task\_progress.

Referenced by MC\_PrintAgentCode\_chdl().

### 13.38.20.3 EXPORTMC int MC\_RegisterService (MCAgency\_t agency, MCAgent\_t agent, int agentID, const char \* agentName, char \*\* serviceNames, int numServices)

Register a new service with the Directory Facilitator.

#### Parameters:

*agency* a MobileC agency handle  
*agent* (OPTIONAL: See note) a MobileC agent  
*agentID* (OPTIONAL: See note) a MobileC agent id  
*agentName* (OPTIONAL: See note) a MobileC agent name  
*serviceNames* an array of character strings of service names  
*numServices* the number of services described in 'serviceNames'

#### Returns:

0 on success, error\_code\_t type on failure

#### Note:

Three of the input arguments are optional. The function expects as input the arguments 'agent XOR (agentID AND agentName)'.

## 13.38.21 Example

Definition at line 1486 of file libmc.c.

References CHECK\_NULL, mc\_platform\_s::df, df\_AddRequest(), df\_request\_list\_node\_New(), agent\_s::id, MC\_ERR\_INVALID\_ARGS, MC\_ERR\_MEMORY, agency\_s::mc\_platform, MUTEX\_INIT, MUTEX\_T, and agent\_s::name.

Referenced by MC\_RegisterService\_chdl().

### 13.38.21.1 EXPORTMC int MC\_ResetSignal (MCAgency\_t attr)

Reset a MobileC signal.

**Returns:**

0 on success, error\_code\_t on failure

**See also:**

[MC\\_WaitSignal\(\)](#)

Definition at line 1618 of file libmc.c.

References COND\_SIGNAL, mc\_platform\_s::giant, mc\_platform\_s::giant\_cond, mc\_platform\_s::giant\_lock, MC\_NO\_SIGNAL, agency\_s::mc\_platform, mc\_platform\_s::MC\_signal, MUTEX\_LOCK, and MUTEX\_UNLOCK.

### 13.38.21.2 EXPORTMC int MC\_ResumeAgency (MCAgency\_t agency)

Resumes a halted agency.

**Parameters:**

*agency* An agency previously halted with the [MC\\_HaltAgency\(\)](#) function.

**Returns:**

0 on success, non-zero on failure.

Definition at line 1565 of file libmc.c.

References mc\_platform\_s::giant, mc\_platform\_s::giant\_lock, agency\_s::mc\_platform, MUTEX\_LOCK, and MUTEX\_UNLOCK.

Referenced by MC\_GetAllAgents(), and MC\_ResumeAgency\_chdl().

### 13.38.21.3 EXPORTMC MCAgent\_t MC\_RetrieveAgent (MCAgency\_t attr)

Retrieves the oldest agent from an agency.

**Returns:**

a valid agent or NULL on failure

Definition at line 1574 of file libmc.c.

References mc\_platform\_s::agent\_queue, agent\_s::agent\_status, ListSearch(), MC\_AGENT\_NEUTRAL, agency\_s::mc\_platform, MUTEX\_LOCK, and MUTEX\_UNLOCK.

Referenced by MC\_RetrieveAgent\_chdl().

### 13.38.21.4 EXPORTMC char\* MC\_RetrieveAgentCode (MCAgent\_t agent)

Retrieves an agent's Ch code.



**Returns:**

a malloc'd character string on success, NULL on failure

Definition at line 1602 of file libmc.c.

References `agent_datastate_s::agent_code`, `buf`, `agent_s::datastate`, `agent_s::lock`, `MUTEX_LOCK`, `MUTEX_UNLOCK`, and `agent_datastate_s::task_progress`.

Referenced by `MC_RetrieveAgentCode_chdl()`.

### 13.38.21.5 EXPORTMC int MC\_SearchForService (MCAgency\_t *attr*, const char \* *searchString*, char \*\*\* *agentNames*, char \*\*\* *serviceNames*, int \*\* *agentIDs*, int \* *numResults*)

Search the directory facilitator for a service.

**Returns:**

0 on success, `error_code_t` on failure

**Parameters:**

*attr* (input) a MobileC agency handle

*searchString* (input) substring to search services for

*agentNames* (return) array of agent names with matching services

*serviceNames* (return) array of matching service names

*agentIDs* (return) array of matching agent IDs

*numResults* (return) number of matching results

## 13.38.22 Example

Definition at line 1629 of file libmc.c.

References `CHECK_NULL`, `COND_SLEEP_ACTION`, `mc_platform_s::df`, `df_AddRequest()`, `df_request_list_node_Destroy()`, `df_request_list_node_New()`, `df_request_search_Destroy()`, `df_request_search_New()`, `MC_ERR_MEMORY`, `agency_s::mc_platform`, `MC_SUCCESS`, and `search`.

Referenced by `MC_SearchForService_chdl()`.

### 13.38.22.1 EXPORTMC int MC\_SemaphorePost (MCAgency\_t *attr*, int *id*)

Post to a MobileC synchronization variable semaphore.

**Parameters:**

*attr* a MobileC agency handle

*id* the synchronization variable id to post to

**Returns:**

0 on success, `error_code_t` type on failure

### 13.38.23 Example

Definition at line 1679 of file libmc.c.

References `agency_s::mc_platform`, `syncListNode_s::sem`, `SEMAPHORE_POST`, `mc_platform_s::syncList`, and `syncListFind()`.

Referenced by `MC_SemaphorePost_chdl()`.

#### 13.38.23.1 EXPORTMC int MC\_SemaphoreWait (MCAgency\_t *attr*, int *id*)

Decreases a MobileC synchronization variable semaphore count by one.

##### Parameters:

*attr* a MobileC agency handle

*id* synchronization variable id to wait on

##### Returns:

0 on MC\_SUCCESS, error\_code\_t type of failure

##### Note:

If the semaphore count is already zero, this function will block until another thread posts to the semaphore.

### 13.38.24 Example

Definition at line 1691 of file libmc.c.

References `agency_s::mc_platform`, `syncListNode_s::sem`, `SEMAPHORE_WAIT`, `mc_platform_s::syncList`, and `syncListFind()`.

Referenced by `MC_SemaphoreWait_chdl()`.

#### 13.38.24.1 EXPORTMC int MC\_SendAgentMigrationMessage (MCAgency\_t *attr*, const char \* *message*, const char \* *hostname*, int *port*)

Sends an agent migration message.

##### Parameters:

*attr* a MobileC agency handle

*message* a valid MobileC xml agent migration message

*hostname* host to send the message to

*port* port to send the message to

Definition at line 1713 of file libmc.c.

References `MC_ERR`, `agency_s::mc_platform`, `message_Destroy()`, `message_InitializeFromString()`, `message_New()`, and `mc_platform_s::message_queue`.

Referenced by `MC_SendAgentMigrationMessage_chdl()`.

**13.38.24.2 EXPORTMC int MC\_SendAgentMigrationMessageFile (MCAgency\_t *attr*, const char \**filename*, const char \**hostname*, int *port*)**

Sends an agent migration message.

**Parameters:**

*attr* a MobileC agency handle  
*filename* file containing a valid MobileC xml agent migration message  
*hostname* hostname to send the agent to  
*port* port to send the agent to

Definition at line 1744 of file libmc.c.

References agent\_Initialize(), mc\_platform\_s::agent\_queue, agent\_s::agent\_status, mc\_platform\_s::ams, buf, COND\_BROADCAST, agency\_s::mc\_platform, MC\_WAIT\_MESSGSEND, message\_Destroy(), message\_InitializeFromString(), message\_New(), MUTEX\_LOCK, and MUTEX\_UNLOCK.

Referenced by handler\_SEND(), and MC\_SendAgentMigrationMessageFile\_chdl().

**13.38.24.3 EXPORTMC int MC\_SetAgentStatus (MCAgent\_t *agent*, int *status*)**

Set an agent's status.

**Parameters:**

*agent* a MobileC agent  
*status* agent status of type 'enum MC\_AgentStatus\_e'

**Returns:**

0 on success, or error\_code\_t on failure

Definition at line 1822 of file libmc.c.

References agent\_s::agent\_status, mc\_platform\_s::ams, COND\_SIGNAL, agent\_s::lock, agent\_s::mc\_platform, MUTEX\_LOCK, MUTEX\_UNLOCK, and agent\_s::orphan.

Referenced by MC\_DeleteAgent(), MC\_DeleteAgentWG(), and MC\_SetAgentStatus\_chdl().

**13.38.24.4 EXPORTMC int MC\_SetDefaultAgentStatus (MCAgency\_t *agency*, enum MC\_AgentStatus\_e *status*)**

Sets default incoming agent status.

**Parameters:**

*agency* a MobileC agency handle  
*status* the status to set all incoming agents

**Returns:**

0 on success, error\_type\_t on failure

**Note:**

using this function will override any status the incoming agent attempts to set for itself.

Definition at line 1837 of file libmc.c.

References mc\_platform\_s::default\_agentstatus, and agency\_s::mc\_platform.

Referenced by MC\_SetDefaultAgentStatus\_chdl().

#### 13.38.24.5 EXPORTMC int MC\_SetThreadOff (MCAgencyOptions\_t \* *options*, enum MC\_ThreadIndex\_e *index*)

Sets a MobileC thread to "off" status.

##### Parameters:

*options* MobileC *options* previously initialized with MC\_InitializeAgencyOptions()

*index* the thread to set

##### Returns:

0 on success, error\_code\_t on failure

##### Note:

This function must be called before MC\_Initialize(). Once an agency is started with MC\_Initialize, the MC\_SetThread functions will have no effect.

Definition at line 1864 of file libmc.c.

References SET\_THREAD\_OFF, and MCAgencyOptions\_s::threads.

#### 13.38.24.6 EXPORTMC int MC\_SetThreadOn (MCAgencyOptions\_t \* *options*, enum MC\_ThreadIndex\_e *index*)

Sets a MobileC thread to "on" status.

##### Parameters:

*options* MobileC *options* previously initialized with MC\_InitializeAgencyOptions()

*index* the thread to set

##### Returns:

0 on success, error\_code\_t on failure

##### Note:

This function must be called before MC\_Initialize(). Once an agency is started with MC\_Initialize, the MC\_SetThread functions will have no effect.

Definition at line 1847 of file libmc.c.

References SET\_THREAD\_ON, and MCAgencyOptions\_s::threads.

#### 13.38.24.7 EXPORTMC int MC\_SetThreadsAllOff (MCAgencyOptions\_t \* *options*)

Set all MobileC threads to 'off' status.

**Parameters:**

*options* a MobileC *options* structure initialized with with the `MC_InitializeAgencyOptions()` function.

**Returns:**

0 on success, error code on failure.

Definition at line 1871 of file libmc.c.

References `MC_THREAD_ALL`, `SET_THREAD_OFF`, and `MCAgencyOptions_s::threads`.

**13.38.24.8 EXPORTMC int MC\_SetThreadsAllOn (MCAgencyOptions\_t \* *options*)**

Set all Mobile-C threads on.

**Parameters:**

*options* MobileC *options* structure, initialized with `MC_InitializeAgencyOptions()`

**Returns:**

0 on success, error code on failure.

Definition at line 1854 of file libmc.c.

References `MC_THREAD_ALL`, `SET_THREAD_ON`, and `MCAgencyOptions_s::threads`.

**13.38.24.9 EXPORTMC int MC\_Steer (MCAgency\_t *attr*, int(\*)(void \**data*) *funcptr*, void \* *arg*)**

Set up a steerable algorithm.

**Parameters:**

*attr* a MobileC agency handle

*funcptr* a function pointer to the algorithm

*arg* an argument for the algorithm function

**Returns:**

0 on success, `error_code_t` on failure

**Note:**

The algorithm function must contain a call to `MC_SteerControl` in order for the algorithm to be steerable.

**13.38.25 Example**

Definition at line 1881 of file libmc.c.

References `agency_s::mc_platform`, `MC_RESTART`, `MC_RUN`, `mc_platform_s::MC_steer_command`, `mc_platform_s::MC_steer_lock`, `MUTEX_LOCK`, and `MUTEX_UNLOCK`.

**13.38.25.1 EXPORTMC enum MC\_SteerCommand\_e MC\_SteerControl (void)**

The MobileC user-algorithm steering function.

**Returns:**

The current steering command

**Note:**

This function belongs inside a user's steerable algorithm.

**See also:**

[MC\\_Steer\(\)](#)

Definition at line 1900 of file libmc.c.

References COND\_WAIT, mc\_platform\_s::MC\_steer\_command, mc\_platform\_s::MC\_steer\_cond, mc\_platform\_s::MC\_steer\_lock, MC\_SUSPEND, MUTEX\_LOCK, and MUTEX\_UNLOCK.

**13.38.25.2 EXPORTMC int MC\_SyncDelete (MCAgency\_t attr, int id)**

Deletes a previously initialized synchronization variable.

**Parameters:**

*attr* a MobileC agency handle

*id* the sync variable id to delete

**Returns:**

0 on success, or error\_code\_t on failure

Definition at line 1916 of file libmc.c.

References syncList\_s::giant\_lock, syncListNode\_s::lock, MC\_ERR\_NOT\_FOUND, agency\_s::mc\_platform, MUTEX\_LOCK, MUTEX\_UNLOCK, mc\_platform\_s::syncList, syncListFind(), syncListNodeDestroy(), and syncListRemove().

Referenced by MC\_SyncDelete\_chdl().

**13.38.25.3 EXPORTMC int MC\_SyncInit (MCAgency\_t attr, int id)**

Initializes a new MobileC synchronization variable.

**Parameters:**

*attr* a MobileC agency handle

*id* the requested sync variable id

**Returns:**

new sync variable's id. May be different than the requested id.

**Note:**

Each synchronization variable may be used as a mutex, condition variable, or semaphore. However, it should only be used as one type of synchronization variable per instance, or undefined behaviour may result.

**See also:**

[MC\\_MutexLock\(\)](#), [MC\\_MutexUnlock\(\)](#), [MC\\_CondWait\(\)](#), [MC\\_CondSignal\(\)](#), [MC\\_CondBroadcast](#), [MC\\_SemaphorePost\(\)](#), [MC\\_SemaphoreWait\(\)](#)

Definition at line 1946 of file libmc.c.

References `syncList_s::giant_lock`, `syncListNode_s::id`, `agency_s::mc_platform`, `MUTEX_LOCK`, `MUTEX_UNLOCK`, `node`, `mc_platform_s::syncList`, `syncListAddNode()`, `syncListFind()`, and `syncListNodeNew()`.

Referenced by `MC_SyncInit_chdl()`.

**13.38.25.4 EXPORTMC int MC\_TerminateAgent (MCAgent\_t *agent*)**

Halt a running agent.

**Returns:**

0 on success, `error_code_t` on failure

Definition at line 1971 of file libmc.c.

References `agent_s::agent_interp`.

Referenced by `ams_ManageAgentList()`, `MC_DeleteAgent()`, and `MC_TerminateAgent_chdl()`.

**13.38.25.5 EXPORTMC int MC\_TerminateAgentWG (MCAgent\_t *calling\_agent*, MCAgent\_t *agent*)**

Definition at line 1981 of file libmc.c.

References `agent_s::agent_interp`, `MC_ERR_INVALID_ARGS`, and `agent_s::wg_code`.

Referenced by `MC_DeleteAgentWG()`, and `MC_TerminateAgentWG_chdl()`.

**13.38.25.6 EXPORTMC int MC\_WaitAgent (MCAgency\_t *attr*)**

Wait indefinitely.

**Note:**

This function is intended to block the calling thread forever. Wait for an agent arrival event  
This function blocks until an agent arrival signal is triggered, at which point it unblocks.

Definition at line 2020 of file libmc.c.

References `mc_platform_s::agent_queue`, `COND_WAIT`, `mc_platform`, `agency_s::mc_platform`, `MUTEX_LOCK`, `MUTEX_UNLOCK`, and `size`.

### 13.38.25.7 EXPORTMC MCAgent\_t MC\_WaitRetrieveAgent (MCAgency\_t attr)

Wait and retrieve an agent.

**Returns:**

a valid MobileC agent on success, or NULL on failure

**Note:**

This function blocks until the arrival of an agent. The agent is retrieved after it is initialized, but before it is executed.

Definition at line 2040 of file libmc.c.

References mc\_platform\_s::agent\_queue, ListSearch(), agency\_s::mc\_platform, MC\_RECV\_AGENT, MC\_WaitSignal(), MUTEX\_LOCK, and MUTEX\_UNLOCK.

### 13.38.25.8 EXPORTMC int MC\_WaitSignal (MCAgency\_t attr, int signals)

Wait for a MobileC signal.

**Parameters:**

*attr* a MobileC agency handle

*signals* a flag of signals to wait for, of type 'enum MC\_Signal\_e'

**Returns:**

0 on success, error\_code\_t on failure

**Note:**

the parameter 'signals' may be something like 'MC\_RECV\_MESSAGE | MC\_RECV\_AGENT', etc.

### 13.38.26 Example

Definition at line 2058 of file libmc.c.

References COND\_WAIT, mc\_platform\_s::giant, mc\_platform\_s::giant\_lock, agency\_s::mc\_platform, mc\_platform\_s::MC\_signal, mc\_platform\_s::MC\_signal\_cond, mc\_platform\_s::MC\_signal\_lock, MUTEX\_LOCK, and MUTEX\_UNLOCK.

Referenced by MC\_WaitRetrieveAgent().



## 13.39 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/macros.h File Reference

```
#include <pthread.h>
#include <semaphore.h>
#include "config.h"
#include <errno.h>
```

### Defines

- #define [GET\\_THREAD\\_MODE](#)(a, b) ( (a & (1<<b)) / (1<<b) )
- #define [SET\\_THREAD\\_ON](#)(a, b) a = (a | (1<<b))
- #define [SET\\_THREAD\\_OFF](#)(a, b) a = (a & (~(1<<b)))
- #define [STRUCT](#)(name, members)
- #define [SOCKET\\_ERROR](#)()
- #define [PTHREAD\\_STACK\\_SIZE](#) 131072
- #define [THREAD\\_CREATE](#)(thread\_handle, function, arg)
- #define [THREAD\\_CANCEL](#)(thread\_handle) pthread\_cancel( thread\_handle )
- #define [THREAD\\_JOIN](#)(thread\_handle) pthread\_join( thread\_handle, NULL )
- #define [THREAD\\_DETACH](#)(thread\_handle)
- #define [THREAD\\_EXIT](#)() pthread\_exit(NULL)
- #define [MUTEX\\_T](#) pthread\_mutex\_t
- #define [MUTEX\\_INIT](#)(mutex) pthread\_mutex\_init(mutex, NULL)
- #define [MUTEX\\_DESTROY](#)(mutex) pthread\_mutex\_destroy(mutex)
- #define [MUTEX\\_LOCK](#)(mutex)
- #define [MUTEX\\_UNLOCK](#)(mutex) pthread\_mutex\_unlock( mutex )
- #define [MUTEX\\_NEW](#)(mutex)
- #define [COND\\_T](#) pthread\_cond\_t
- #define [COND\\_INIT](#)(cond) pthread\_cond\_init([cond](#), NULL)
- #define [COND\\_DESTROY](#)(cond) pthread\_cond\_destroy([cond](#))
- #define [COND\\_WAIT](#)(cond, mutex) pthread\_cond\_wait([cond](#), mutex )
- #define [COND\\_SLEEP](#)(cond, mutex, test)
- #define [COND\\_RESET](#)(cond, mutex) pthread\_mutex\_unlock( mutex );
- #define [COND\\_SLEEP\\_ACTION](#)(cond, mutex, action)
- #define [SIGNAL](#)(cond, mutex, action)
- #define [COND\\_BROADCAST](#)(cond) pthread\_cond\_broadcast( [cond](#) )
- #define [COND\\_SIGNAL](#)(cond) pthread\_cond\_signal( [cond](#) )
- #define [SEMAPHORE\\_T](#) sem\_t
- #define [SEMAPHORE\\_INIT](#)(sem) sem\_init(sem, 0, 0)
- #define [SEMAPHORE\\_DESTROY](#)(sem) sem\_destroy(sem)
- #define [SEMAPHORE\\_WAIT](#)(sem) sem\_wait(sem)
- #define [SEMAPHORE\\_POST](#)(sem) sem\_post(sem)
- #define [RWLOCK\\_T](#) mc\_rwlock\_t
- #define [RWLOCK\\_INIT](#)(rwlock) mc\_rwlock\_init(rwlock)
- #define [RWLOCK\\_DESTROY](#)(rwlock) mc\_rwlock\_destroy(rwlock)
- #define [RWLOCK\\_RDLOCK](#)(rwlock) mc\_rwlock\_rdlock(rwlock)
- #define [RWLOCK\\_RDUNLOCK](#)(rwlock) mc\_rwlock\_rdunlock(rwlock)
- #define [RWLOCK\\_WRLOCK](#)(rwlock) mc\_rwlock\_wrlock(rwlock)

- #define [RWLOCK\\_WRUNLOCK](#)(rwlock) mc\_rwlock\_wrunlock(rwlock)
- #define [WAKE\\_QUEUE](#)(queue, action)
- #define [SLEEP\\_QUEUE](#)(queue)
- #define [SLEEP\\_RESET](#)(queue) pthread\_mutex\_unlock( queue->thread\_mutex )
- #define [CHECK\\_NULL](#)(var, action)
- #define [WARN](#)(message)
- #define [CH\\_DATATYPE\\_SIZE](#)(type, size)
- #define [CH\\_DATATYPE\\_STRING](#)(type, string)
- #define [CH\\_DATATYPE\\_VALUE\\_STRING](#)(type, string, p)
- #define [CH\\_STRING\\_DATATYPE](#)(string, type)
- #define [CH\\_DATATYPE\\_STR\\_TO\\_VAL](#)(type, string, val)

### 13.39.1 Define Documentation

#### 13.39.1.1 #define CH\_DATATYPE\_SIZE(type, size)

**Value:**

```

switch(type) {
    case CH_CHARTYPE:
        size = sizeof(char);
        break;
    case CH_INTTYPE:
        size = sizeof(int);
        break;
    case CH_UINTTYPE:
        size = sizeof(unsigned int);
        break;
    case CH_SHORTTYPE:
        size = sizeof(short);
        break;
    case CH_USHORTTYPE:
        size = sizeof(unsigned short);
        break;
    case CH_FLOATTYPE:
        size = sizeof(float);
        break;
    case CH_DOUBLETTYPE:
        size = sizeof(double);
        break;
    default:
        fprintf(stderr, "Unknown data type: %d at %s:%d",
            type, __FILE__, __LINE__);
        size=0;
}

```

Definition at line 491 of file macros.h.

Referenced by [agent\\_AddPersistentVariable\(\)](#), [agent\\_xml\\_compose\\_\\_create\\_row\\_nodes\(\)](#), [agent\\_xml\\_parse\\_\\_data\(\)](#), [agent\\_xml\\_parse\\_\\_fill\\_row\\_data\(\)](#), [agent\\_xml\\_parse\\_\\_row\(\)](#), [interpreter\\_variable\\_data\\_Initialize\(\)](#), [interpreter\\_variable\\_data\\_InitializeFromAgent\(\)](#), and [MC\\_GetAgentReturnData\(\)](#).

#### 13.39.1.2 #define CH\_DATATYPE\_STR\_TO\_VAL(type, string, val)

**Value:**

```

switch (type) { \
    case CH_INTTYPE: \

```

```

        *(int*)val = atoi(string); \
break; \
case CH_UINTTYPE: \
        *(unsigned int*)val = atoi(string); \
break; \
case CH_SHORTTYPE: \
        *(short*)val = (short)atoi(string); /*FIXME*/ \
break; \
case CH_USHORTTYPE: \
        *(unsigned short*)val = (unsigned short)atoi(string); /*F
        IXME*/ \
break; \
case CH_FLOATTYPE: \
        *(float*)val = strtod(string, NULL); \
break; \
case CH_DOUBLETTYPE: \
        *(double*)val = strtod(string, NULL); \
break; \
default: \
        fprintf(stderr, \
                "Unsupported data type: %d %s:%d\n", \
                type, __FILE__, __LINE__ ); \
}

```

Definition at line 605 of file macros.h.

Referenced by agent\_xml\_parse\_\_data().

### 13.39.1.3 #define CH\_DATATYPE\_STRING(type, string)

**Value:**

```

switch(type) { \
    case CH_CHARTYPE: \
        strcpy(string, "char"); \
    break; \
    case CH_INTTYPE: \
        strcpy(string, "int"); \
    break; \
    case CH_UINTTYPE: \
        strcpy(string, "unsigned int"); \
    break; \
    case CH_SHORTTYPE: \
        strcpy(string, "short"); \
    break; \
    case CH_USHORTTYPE: \
        strcpy(string, "unsigned short"); \
    break; \
    case CH_FLOATTYPE: \
        strcpy(string, "float"); \
    break; \
    case CH_DOUBLETTYPE: \
        strcpy(string, "double"); \
    break; \
    default: \
        fprintf(stderr, \
                "Unsupported data type: %d %s:%d\n", \
                type, __FILE__, __LINE__ ); \
}

```

Definition at line 522 of file macros.h.

Referenced by agent\_xml\_compose\_\_data().

### 13.39.1.4 #define CH\_DATATYPE\_VALUE\_STRING(type, string, p)

**Value:**

```

switch(type) {
    case CH_CHARTYPE:
        sprintf(string, "%c", *((char*)p));
        break;
    case CH_INTTYPE:
        sprintf(string, "%d", *((int*)p));
        break;
    case CH_UINTTYPE:
        sprintf(string, "%d", *((unsigned int*)p));
        break;
    case CH_SHORTTYPE:
        sprintf(string, "%d", *((short*)p));
        break;
    case CH_USHORTTYPE:
        sprintf(string, "%d", *((unsigned short*)p));
        break;
    case CH_FLOATTYPE:
        sprintf(string, "%f", *((float*)p));
        break;
    case CH_DOUBLETTYPE:
        sprintf(string, "%f", *((double*)p));
        break;
    default:
        fprintf(stderr,
            "Unsupported data type: %d %s:%d\n",
            type, __FILE__, __LINE__);
}

```

Definition at line 554 of file macros.h.

Referenced by agent\_xml\_compose\_\_create\_row\_nodes(), and agent\_xml\_compose\_\_data().

### 13.39.1.5 #define CH\_STRING\_DATATYPE(string, type)

**Value:**

```

if (!strcmp(string, "int")) {
    type = CH_INTTYPE;
} else if (!strcmp(string, "float")) {
    type = CH_FLOATTYPE;
} else if (!strcmp(string, "double")) {
    type = CH_DOUBLETTYPE;
} else if (!strcmp(string, "unsigned int")) {
    type = CH_UINTTYPE;
} else if (!strcmp(string, "short")) {
    type = CH_SHORTTYPE;
} else if (!strcmp(string, "unsigned short")) {
    type = CH_USHORTTYPE;
} else if (!strcmp(string, "char")) {
    type = CH_CHARTYPE;
} else {
    fprintf(stderr,
        "Unsupported data type: %d %s:%d\n",
        type, __FILE__, __LINE__);
}

```

Definition at line 583 of file macros.h.

Referenced by agent\_xml\_parse\_\_data().

**13.39.1.6 #define CHECK\_NULL(var, action)****Value:**

```

if ( var == NULL ) {
    fprintf(stderr, "Pointer var is null: expected otherwise.\n");
    fprintf(stderr, "Error occured at %s:%d", __FILE__, __LINE__);
    action;
}

```

Definition at line 474 of file macros.h.

Referenced by agent\_AddPersistentVariable(), agent\_datastate\_New(), agent\_xml\_parse\_home(), agent\_xml\_parse\_name(), agent\_xml\_parse\_owner(), agent\_xml\_parse\_sender(), agent\_xml\_parse\_task(), ams\_Initialize(), barrier\_node\_Initialize(), barrier\_queue\_New(), df\_request\_list\_New(), df\_request\_list\_node\_New(), df\_request\_search\_New(), fipa\_agent\_identifier\_Parse(), fipa\_word\_Parse(), http\_ParseExpression(), interpreter\_variable\_data\_Initialize(), interpreter\_variable\_data\_InitializeFromAgent(), interpreter\_variable\_data\_New(), MC\_AclSend\_chdl(), MC\_AddAgent\_chdl(), MC\_Barrier\_chdl(), MC\_BarrierDelete\_chdl(), MC\_BarrierInit\_chdl(), MC\_CondBroadcast\_chdl(), MC\_CondReset\_chdl(), MC\_CondSignal\_chdl(), MC\_CondWait\_chdl(), MC\_DeleteAgent(), MC\_DeleteAgentWG(), MC\_DeregisterService\_chdl(), MC\_End\_chdl(), MC\_FindAgentByID\_chdl(), MC\_FindAgentByName\_chdl(), MC\_HaltAgency\_chdl(), MC\_Initialize(), MC\_MutexLock\_chdl(), MC\_MutexUnlock\_chdl(), mc\_platform\_Initialize(), MC\_RegisterService(), MC\_RegisterService\_chdl(), MC\_ResumeAgency\_chdl(), MC\_RetrieveAgent\_chdl(), mc\_rwlock\_init(), MC\_SearchForService(), MC\_SearchForService\_chdl(), MC\_SemaphorePost\_chdl(), MC\_SemaphoreWait\_chdl(), MC\_SendAgentMigrationMessage\_chdl(), MC\_SendSteerCommand\_chdl(), MC\_SetDefaultAgentStatus\_chdl(), MC\_SyncDelete\_chdl(), MC\_SyncInit\_chdl(), message\_InitializeFromAgent(), message\_InitializeFromConnection(), message\_InitializeFromString(), message\_New(), message\_queue\_SendOutgoing(), message\_send\_Thread(), message\_xml\_parse\_message(), mtp\_http\_InitializeFromConnection(), mtp\_http\_New(), syncListNodeInit(), syncListNodeNew(), xml\_get\_cdata(), xml\_get\_text(), and xml\_new\_cdata().

**13.39.1.7 #define COND\_BROADCAST(cond) pthread\_cond\_broadcast( cond )**

Definition at line 226 of file macros.h.

Referenced by acc\_MessageHandlerThread(), acc\_Thread(), ams\_Thread(), df\_Thread(), handler\_QUIT(), listen\_Thread(), MC\_Barrier(), MC\_CondBroadcast(), MC\_End(), MC\_SendAgentMigrationMessageFile(), and MC\_SendSteerCommand().

**13.39.1.8 #define COND\_DESTROY(cond) pthread\_cond\_destroy(cond)**

Definition at line 202 of file macros.h.

Referenced by ams\_Destroy(), barrier\_node\_Destroy(), df\_Destroy(), df\_request\_list\_node\_Destroy(), df\_request\_search\_Destroy(), mc\_platform\_Destroy(), mc\_rwlock\_destroy(), and syncListNodeDestroy().

**13.39.1.9 #define COND\_INIT(cond) pthread\_cond\_init(cond, NULL)**

Definition at line 199 of file macros.h.

Referenced by acc\_Initialize(), ams\_Initialize(), barrier\_node\_Initialize(), df\_Initialize(), df\_request\_list\_New(), df\_request\_list\_node\_New(), df\_request\_search\_New(), mc\_platform\_Initialize(), mc\_rwlock\_init(), syncListNodeInit(), and syncListNodeNew().

**13.39.1.10 #define COND\_RESET(cond, mutex) pthread\_mutex\_unlock( mutex );**

Definition at line 214 of file macros.h.

**13.39.1.11 #define COND\_SIGNAL(cond) pthread\_cond\_signal( cond )**

Definition at line 228 of file macros.h.

Referenced by agent\_RunChScriptThread(), AP\_QUEUE\_STD\_DEFN\_TEMPLATE(), MC\_AddAgent(), MC\_CondSignal(), MC\_End(), MC\_ResetSignal(), mc\_rwlock\_rdunlock(), mc\_rwlock\_wrunlock(), and MC\_SetAgentStatus().

**13.39.1.12 #define COND\_SLEEP(cond, mutex, test)**

**Value:**

```
if (pthread_mutex_lock( mutex )) \
    printf("pthread lock error: %s:%d\n", __FILE__, __LINE__); \
if (!test) { \
    pthread_cond_wait( cond, mutex ); \
}
```

Definition at line 208 of file macros.h.

**13.39.1.13 #define COND\_SLEEP\_ACTION(cond, mutex, action)**

**Value:**

```
if (pthread_mutex_lock( mutex )) \
    printf("pthread lock error: %s:%d\n", __FILE__, __LINE__); \
action; \
pthread_cond_wait( cond, mutex );
```

Definition at line 216 of file macros.h.

Referenced by MC\_SearchForService().

**13.39.1.14 #define COND\_T pthread\_cond\_t**

Definition at line 197 of file macros.h.

Referenced by acc\_Initialize(), ams\_Initialize(), barrier\_node\_Initialize(), df\_Initialize(), df\_request\_list\_New(), df\_request\_list\_node\_New(), df\_request\_search\_New(), mc\_platform\_Initialize(), mc\_rwlock\_init(), syncListNodeInit(), and syncListNodeNew().

**13.39.1.15 #define COND\_WAIT(cond, mutex) pthread\_cond\_wait(cond, mutex )**

Definition at line 205 of file macros.h.

Referenced by acc\_MessageHandlerThread(), acc\_Thread(), agent\_mailbox\_WaitRetrieve(), ams\_Thread(), df\_Thread(), MC\_Barrier(), MC\_CondWait(), MC\_MainLoop(), mc\_platform\_Initialize(), mc\_rwlock\_rdlock(), mc\_rwlock\_wrlock(), MC\_SterControl(), MC\_WaitAgent(), MC\_WaitSignal(), and message\_Send().

**13.39.1.16 #define GET\_THREAD\_MODE(a, b) ( (a & (1<<b)) / (1<<b) )**

Definition at line 112 of file macros.h.

Referenced by MC\_End(), and mc\_platform\_Initialize().

**13.39.1.17 #define MUTEX\_DESTROY(mutex) pthread\_mutex\_destroy(mutex)**

Definition at line 180 of file macros.h.

Referenced by agent\_Destroy(), agent\_Initialize(), ams\_Destroy(), barrier\_node\_Destroy(), df\_Destroy(), df\_request\_list\_node\_Destroy(), df\_request\_search\_Destroy(), mc\_platform\_Destroy(), mc\_rwlock\_destroy(), and syncListNodeDestroy().

**13.39.1.18 #define MUTEX\_INIT(mutex) pthread\_mutex\_init(mutex, NULL)**

Definition at line 177 of file macros.h.

Referenced by acc\_Initialize(), agent\_Copy(), agent\_Initialize(), agent\_New(), agent\_NewBinary(), ams\_Initialize(), barrier\_node\_Initialize(), df\_Initialize(), df\_request\_list\_New(), df\_request\_list\_node\_New(), df\_request\_search\_New(), mc\_platform\_Initialize(), MC\_RegisterService(), mc\_rwlock\_init(), syncListInit(), syncListNodeInit(), and syncListNodeNew().

**13.39.1.19 #define MUTEX\_LOCK(mutex)**

**Value:**

```
if (pthread_mutex_lock( mutex )) \
    fprintf(stderr, "pthread lock error: %s:%d\n", __FILE__, __LINE__)
```

Definition at line 183 of file macros.h.

Referenced by acc\_MessageHandlerThread(), acc\_Thread(), agent\_Copy(), agent\_Destroy(), agent\_mailbox\_WaitRetrieve(), agent\_queue\_Flush(), agent\_RunChScriptThread(), ams\_ManageAgentList(), ams\_Print(), ams\_Thread(), AP\_QUEUE\_SEARCH\_TEMPLATE(), AP\_QUEUE\_STD\_DEFN\_TEMPLATE(), df\_Destroy(), df\_node\_Destroy(), df\_request\_list\_Pop(), df\_SearchForService(), df\_Thread(), handler\_QUIT(), interpreter\_variable\_data\_Initialize(), listen\_Thread(), MC\_AddAgent(), MC\_Barrier(), MC\_CallAgentFunc(), MC\_CallAgentFuncArg(), MC\_CallAgentFuncV(), MC\_CallAgentFuncVar(), MC\_CondBroadcast(), MC\_CondReset(), MC\_CondSignal(), MC\_CondWait(), MC\_End(), MC\_GetAgentName(), MC\_GetAgentStatus(), MC\_GetAllAgents(), MC\_HaltAgency(), MC\_MainLoop(), MC\_MutexLock(), mc\_platform\_Initialize(), MC\_PrintAgentCode(), MC\_ResetSignal(), MC\_ResumeAgency(), MC\_RetrieveAgent(), MC\_RetrieveAgentCode(), mc\_rwlock\_rdlock(), mc\_rwlock\_rdunlock(), mc\_rwlock\_wrlock(), mc\_rwlock\_wrunlock(), MC\_SendAgentMigrationMessageFile(), MC\_SendSteerCommand(), MC\_SetAgentStatus(), MC\_Steer(), MC\_SteerControl(), MC\_SyncDelete(), MC\_SyncInit(), MC\_WaitAgent(), MC\_WaitRetrieveAgent(), MC\_WaitSignal(), message\_queue\_SendOutgoing(), message\_Send(), and request\_handler\_DEREGISTER().

**13.39.1.20 #define MUTEX\_NEW(mutex)**

**Value:**

```
mutex = (pthread_mutex_t*)malloc(sizeof(pthread_mutex_t)); \
    if (mutex == NULL) \
        fprintf(stderr, "Memory Error. %s:%d\n", __FILE__, __LINE__); \
```

Definition at line 188 of file macros.h.

Referenced by agent\_New().

### 13.39.1.21 **#define MUTEX\_T pthread\_mutex\_t**

Definition at line 175 of file macros.h.

Referenced by acc\_Initialize(), agent\_Copy(), agent\_Initialize(), agent\_NewBinary(), ams\_Initialize(), barrier\_node\_Initialize(), df\_Initialize(), df\_request\_list\_New(), df\_request\_list\_node\_New(), df\_request\_search\_New(), mc\_platform\_Initialize(), MC\_RegisterService(), mc\_rwlock\_init(), syncListInit(), syncListNodeInit(), and syncListNodeNew().

### 13.39.1.22 **#define MUTEX\_UNLOCK(mutex) pthread\_mutex\_unlock( mutex )**

Definition at line 186 of file macros.h.

Referenced by acc\_MessageHandlerThread(), acc\_Thread(), agent\_mailbox\_WaitRetrieve(), agent\_queue\_Flush(), agent\_RunChScriptThread(), ams\_ManageAgentList(), ams\_Print(), ams\_Thread(), AP\_QUEUE\_SEARCH\_TEMPLATE(), AP\_QUEUE\_STD\_DEFN\_TEMPLATE(), df\_request\_list\_Pop(), df\_SearchForService(), df\_Thread(), handler\_QUIT(), interpreter\_variable\_data\_Initialize(), listen\_Thread(), MC\_AddAgent(), MC\_Barrier(), MC\_CallAgentFunc(), MC\_CallAgentFuncArg(), MC\_CallAgentFuncV(), MC\_CallAgentFuncVar(), MC\_CondBroadcast(), MC\_CondReset(), MC\_CondSignal(), MC\_CondWait(), MC\_End(), MC\_GetAgentName(), MC\_GetAgentStatus(), MC\_GetAllAgents(), MC\_HaltAgency(), MC\_MainLoop(), MC\_MutexUnlock(), mc\_platform\_Initialize(), MC\_PrintAgentCode(), MC\_ResetSignal(), MC\_ResumeAgency(), MC\_RetrieveAgent(), MC\_RetrieveAgentCode(), mc\_rwlock\_rdlock(), mc\_rwlock\_rdunlock(), mc\_rwlock\_wrlock(), mc\_rwlock\_wrunlock(), MC\_SendAgentMigrationMessageFile(), MC\_SendSteerCommand(), MC\_SetAgentStatus(), MC\_Steer(), MC\_SteerControl(), MC\_SyncDelete(), MC\_SyncInit(), MC\_WaitAgent(), MC\_WaitRetrieveAgent(), MC\_WaitSignal(), message\_queue\_SendOutgoing(), message\_Send(), and request\_handler\_DEREGISTER().

### 13.39.1.23 **#define PTHREAD\_STACK\_SIZE 131072**

Definition at line 139 of file macros.h.

### 13.39.1.24 **#define RWLOCK\_DESTROY(rwlock) mc\_rwlock\_destroy(rwlock)**

Definition at line 270 of file macros.h.

Referenced by barrier\_queue\_Destroy().

### 13.39.1.25 **#define RWLOCK\_INIT(rwlock) mc\_rwlock\_init(rwlock)**

Definition at line 262 of file macros.h.

Referenced by barrier\_queue\_New(), and syncListInit().

### 13.39.1.26 **#define RWLOCK\_RDLOCK(rwlock) mc\_rwlock\_rdlock(rwlock)**

Definition at line 288 of file macros.h.

Referenced by barrier\_queue\_Get(), and syncListFind().



**13.39.1.27 #define RWLOCK\_RDUNLOCK(rwlock) mc\_rwlock\_rdunlock(rwlock)**

Definition at line 290 of file macros.h.

Referenced by barrier\_queue\_Get(), and syncListFind().

**13.39.1.28 #define RWLOCK\_T mc\_rwlock\_t**

Definition at line 255 of file macros.h.

Referenced by barrier\_queue\_New(), and syncListInit().

**13.39.1.29 #define RWLOCK\_WRLOCK(rwlock) mc\_rwlock\_wrlock(rwlock)**

Definition at line 292 of file macros.h.

Referenced by barrier\_queue\_Add(), barrier\_queue\_Delete(), syncListAddNode(), syncListDelete(), and syncListRemove().

**13.39.1.30 #define RWLOCK\_WRUNLOCK(rwlock) mc\_rwlock\_wrunlock(rwlock)**

Definition at line 294 of file macros.h.

Referenced by barrier\_queue\_Add(), barrier\_queue\_Delete(), syncListAddNode(), syncListDelete(), and syncListRemove().

**13.39.1.31 #define SEMAPHORE\_DESTROY(sem) sem\_destroy(sem)**

Definition at line 240 of file macros.h.

Referenced by syncListNodeDestroy().

**13.39.1.32 #define SEMAPHORE\_INIT(sem) sem\_init(sem, 0, 0)**

Definition at line 237 of file macros.h.

Referenced by syncListNodeInit(), and syncListNodeNew().

**13.39.1.33 #define SEMAPHORE\_POST(sem) sem\_post(sem)**

Definition at line 245 of file macros.h.

Referenced by MC\_SemaphorePost().

**13.39.1.34 #define SEMAPHORE\_T sem\_t**

Definition at line 235 of file macros.h.

Referenced by syncListNodeInit(), and syncListNodeNew().

**13.39.1.35 #define SEMAPHORE\_WAIT(sem) sem\_wait(sem)**

Definition at line 243 of file macros.h.

Referenced by MC\_SemaphoreWait().

**13.39.1.36 #define SET\_THREAD\_OFF(a, b) a = (a & ~(1<<b))**

Definition at line 116 of file macros.h.

Referenced by MC\_SetThreadOff(), and MC\_SetThreadsAllOff().

**13.39.1.37 #define SET\_THREAD\_ON(a, b) a = (a | (1<<b))**

Definition at line 115 of file macros.h.

Referenced by MC\_SetThreadOn(), and MC\_SetThreadsAllOn().

**13.39.1.38 #define SIGNAL(cond, mutex, action)**

**Value:**

```
pthread_mutex_lock( mutex ); \
action; \
pthread_cond_signal( cond ); \
pthread_mutex_unlock( mutex )
```

Definition at line 221 of file macros.h.

Referenced by agent\_RunChScriptThread(), df\_Add(), df\_AddRequest(), and request\_handler\_SEARCH().

**13.39.1.39 #define SLEEP\_QUEUE(queue)**

**Value:**

```
if (pthread_mutex_lock( queue->thread_mutex )) \
printf("pthread lock error: %s:%d\n", __FILE__, __LINE__); \
pthread_cond_wait( queue->touched_signal, queue->thread_mutex )
```

Definition at line 308 of file macros.h.

**13.39.1.40 #define SLEEP\_RESET(queue) pthread\_mutex\_unlock( queue->thread\_mutex )**

Definition at line 312 of file macros.h.

**13.39.1.41 #define SOCKET\_ERROR()**

**Value:**

```
printf("Socket error. %s:%d\nerrno:%d", __FILE__, __LINE__, errno); \
sleep(500)
```

Definition at line 132 of file macros.h.

Referenced by listen\_Thread(), mc\_platform\_Destroy(), message\_InitializeFromConnection(), message\_send\_Thread(), mtp\_http\_InitializeFromConnection(), net\_bind(), and net\_connect().

#### 13.39.1.42 #define STRUCT(name, members)

**Value:**

```
typedef struct name##_s { \
    members \
} name##_t; \
typedef name##_t* name##_p;
```

Definition at line 120 of file macros.h.

#### 13.39.1.43 #define THREAD\_CANCEL(thread\_handle) pthread\_cancel( thread\_handle )

Definition at line 156 of file macros.h.

Referenced by MC\_End().

#### 13.39.1.44 #define THREAD\_CREATE(thread\_handle, function, arg)

**Value:**

```
while(pthread_create( \
    thread_handle, \
    &attr, \
    function, \
    (void*) arg \
) < 0) { \
    printf("pthread_create failed. Trying again...\n"); \
    usleep(100000); \
}
```

Definition at line 145 of file macros.h.

Referenced by acc\_Start(), acc\_Thread(), agent\_RunChScript(), ams\_Start(), cmd\_prompt\_Start(), df\_Start(), MC\_AddStationaryAgent(), and message\_Send().

#### 13.39.1.45 #define THREAD\_DETACH(thread\_handle)

**Value:**

```
if(pthread_detach(thread_handle) < 0) { \
    printf("pthread_detach failed. %s:%d\n", __FILE__, __LINE__); \
}
```

Definition at line 162 of file macros.h.

Referenced by acc\_Thread(), and message\_Send().

#### 13.39.1.46 #define THREAD\_EXIT() pthread\_exit(NULL)

Definition at line 167 of file macros.h.

Referenced by `acc_MessageHandlerThread()`, `acc_Thread()`, `ams_Thread()`, `df_Thread()`, and `listen_Thread()`.

#### 13.39.1.47 `#define THREAD_JOIN(thread_handle) pthread_join( thread_handle, NULL )`

Definition at line 159 of file `macros.h`.

Referenced by `MC_End()`.

#### 13.39.1.48 `#define WAKE_QUEUE(queue, action)`

**Value:**

```
if (pthread_mutex_trylock( queue->lock ) == 0) {    \
    action;                                          \
    pthread_cond_signal( queue->cond);              \
    pthread_mutex_unlock( queue->lock);              \
}
```

Definition at line 302 of file `macros.h`.

#### 13.39.1.49 `#define WARN(message)`

**Value:**

```
fprintf(stderr, "WARNING: "); \
    fprintf(stderr, message ); \
    fprintf(stderr, " %s:%d\n", __FILE__, __LINE__ )
```

Definition at line 481 of file `macros.h`.

Referenced by `message_InitializeFromAgent()`.

## 13.40 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/mc\_error.h File Reference

### Typedefs

- typedef enum [error\\_code\\_e](#) [error\\_code\\_t](#)

### Enumerations

- enum [error\\_code\\_e](#) {  
[MC\\_SUCCESS](#) = 0, [MC\\_ERR](#), [MC\\_ERR\\_CONNECT](#), [MC\\_ERR\\_PARSE](#),  
[MC\\_ERR\\_EMPTY](#), [MC\\_ERR\\_INVALID](#), [MC\\_ERR\\_INVALID\\_ARGS](#), [MC\\_ERR\\_NOT\\_FOUND](#),  
[MC\\_ERR\\_MEMORY](#), [MC\\_ERR\\_SEND](#), [MC\\_WARN\\_DUPLICATE](#), [MC\\_SUCCESS](#) = 0,  
[ERR](#), [MC\\_ERR\\_CONNECT](#), [MC\\_ERR\\_PARSE](#), [MC\\_ERR\\_EMPTY](#),  
[MC\\_ERR\\_INVALID](#), [MC\\_ERR\\_INVALID\\_ARGS](#), [MC\\_ERR\\_NOT\\_FOUND](#), [MC\\_ERR\\_MEMORY](#),  
[MC\\_ERR\\_SEND](#), [MC\\_WARN\\_DUPLICATE](#) }

#### 13.40.1 Typedef Documentation

##### 13.40.1.1 typedef enum [error\\_code\\_e](#) [error\\_code\\_t](#)

#### 13.40.2 Enumeration Type Documentation

##### 13.40.2.1 enum [error\\_code\\_e](#)

Enumerator:

*MC\_SUCCESS*  
*MC\_ERR*  
*MC\_ERR\_CONNECT*  
*MC\_ERR\_PARSE*  
*MC\_ERR\_EMPTY*  
*MC\_ERR\_INVALID*  
*MC\_ERR\_INVALID\_ARGS*  
*MC\_ERR\_NOT\_FOUND*  
*MC\_ERR\_MEMORY*  
*MC\_ERR\_SEND*  
*MC\_WARN\_DUPLICATE*  
*MC\_SUCCESS*  
*ERR*  
*MC\_ERR\_CONNECT*  
*MC\_ERR\_PARSE*  
*MC\_ERR\_EMPTY*

*MC\_ERR\_INVALID*  
*MC\_ERR\_INVALID\_ARGS*  
*MC\_ERR\_NOT\_FOUND*  
*MC\_ERR\_MEMORY*  
*MC\_ERR\_SEND*  
*MC\_WARN\_DUPLICATE*

Definition at line 41 of file mc\_error.h.

## 13.41 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/mc\_platform.h File Reference

```
#include "acc.h"
#include "ams.h"
#include "barrier.h"
#include "cmd_prompt.h"
#include "config.h"
#include "connection.h"
#include "data_structures.h"
#include "df.h"
#include "libmc.h"
#include "../mc_sync/sync_list.h"
#include "../security/asm.h"
```

### Data Structures

- struct [mc\\_platform\\_s](#)

### Typedefs

- typedef struct [mc\\_platform\\_s](#) [mc\\_platform\\_t](#)
- typedef [mc\\_platform\\_t](#) \* [mc\\_platform\\_p](#)

### Functions

- [mc\\_platform\\_p](#) [mc\\_platform\\_Initialize](#) ([MCAgency\\_t](#) agency, [ChOptions\\_t](#) \*ch\_options)
- [int](#) [mc\\_platform\\_Destroy](#) ([mc\\_platform\\_p](#) mc\_platform)

#### 13.41.1 Typedef Documentation

##### 13.41.1.1 typedef [mc\\_platform\\_t](#)\* [mc\\_platform\\_p](#)

Definition at line 121 of file [mc\\_platform.h](#).

##### 13.41.1.2 typedef struct [mc\\_platform\\_s](#) [mc\\_platform\\_t](#)

Definition at line 120 of file [mc\\_platform.h](#).

#### 13.41.2 Function Documentation

##### 13.41.2.1 [int](#) [mc\\_platform\\_Destroy](#) ([mc\\_platform\\_p](#) mc\_platform)

Definition at line 244 of file [mc\\_platform.c](#).

References `mc_platform_s::acc`, `acc_Destroy()`, `mc_platform_s::agent_queue`, `mc_platform_s::ams`, `ams_Destroy()`, `mc_platform_s::barrier_queue`, `barrier_queue_Destroy()`, `mc_platform_s::cmd_prompt`, `cmd_prompt_Destroy()`, `COND_DESTROY`, `mc_platform_s::connection_queue`, `mc_platform_s::df`, `df_Destroy()`, `mc_platform_s::giant_cond`, `mc_platform_s::giant_lock`, `mc_platform_s::interp_options`, `mc_platform_s::MC_signal_cond`, `mc_platform_s::MC_signal_lock`, `mc_platform_s::MC_steer_cond`, `mc_platform_s::MC_steer_lock`, `MC_SUCCESS`, `mc_platform_s::MC_sync_cond`, `mc_platform_s::MC_sync_lock`, `mc_platform_s::message_queue`, `MUTEX_DESTROY`, `mc_platform_s::quit_lock`, `SOCKET_ERROR`, and `mc_platform_s::sockfd`.

Referenced by `MC_End()`.

### 13.41.2.2 `mc_platform_p mc_platform_Initialize (MCAgency_t agency, ChOptions_t * ch_options)`

Definition at line 53 of file `mc_platform.c`.

References `acc_Initialize()`, `acc_Start()`, `agent_ChScriptInitVar()`, `ams_Initialize()`, `ams_Start()`, `barrier_queue_New()`, `CHECK_NULL`, `cmd_prompt_Initialize()`, `cmd_prompt_Start()`, `COND_INIT`, `COND_T`, `COND_WAIT`, `agency_s::default_agentstatus`, `DEFAULT_HOSTNAME_LENGTH`, `df_Initialize()`, `df_Start()`, `GET_THREAD_MODE`, `agency_s::initInterps`, `agency_s::last_error`, `MC_ERR_MEMORY`, `MC_NO_SIGNAL`, `mc_platform`, `MC_THREAD_ACC`, `MC_THREAD_ALL`, `MC_THREAD_AMS`, `MC_THREAD_CP`, `MC_THREAD_DF`, `MUTEX_INIT`, `MUTEX_LOCK`, `MUTEX_T`, `MUTEX_UNLOCK`, `agency_s::portno`, `agency_s::stack_size`, `syncListInit()`, and `agency_s::threads`.

Referenced by `MC_Initialize()`.



## 13.42 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/mc\_rwlock.h File Reference

```
#include "macros.h"
```

### Data Structures

- struct [mc\\_rwlock\\_s](#)

### Typedefs

- typedef struct [mc\\_rwlock\\_s](#) [mc\\_rwlock\\_t](#)
- typedef [mc\\_rwlock\\_t](#) \* [mc\\_rwlock\\_p](#)

### Functions

- [int mc\\_rwlock\\_init](#) ([mc\\_rwlock\\_p](#) *rwlock*)
- [int mc\\_rwlock\\_destroy](#) ([mc\\_rwlock\\_p](#) *rwlock*)
- [int mc\\_rwlock\\_rdlock](#) ([mc\\_rwlock\\_p](#) *rwlock*)
- [int mc\\_rwlock\\_rdunlock](#) ([mc\\_rwlock\\_p](#) *rwlock*)
- [int mc\\_rwlock\\_wrlock](#) ([mc\\_rwlock\\_p](#) *rwlock*)
- [int mc\\_rwlock\\_wrunlock](#) ([mc\\_rwlock\\_p](#) *rwlock*)

#### 13.42.1 Typedef Documentation

##### 13.42.1.1 typedef [mc\\_rwlock\\_t](#)\* [mc\\_rwlock\\_p](#)

Definition at line 46 of file [mc\\_rwlock.h](#).

##### 13.42.1.2 typedef struct [mc\\_rwlock\\_s](#) [mc\\_rwlock\\_t](#)

#### 13.42.2 Function Documentation

##### 13.42.2.1 [int mc\\_rwlock\\_destroy](#) ([mc\\_rwlock\\_p](#) *rwlock*)

Definition at line 66 of file [mc\\_rwlock.c](#).

References [mc\\_rwlock\\_s::cond](#), [COND\\_DESTROY](#), [mc\\_rwlock\\_s::lock](#), and [MUTEX\\_DESTROY](#).

##### 13.42.2.2 [int mc\\_rwlock\\_init](#) ([mc\\_rwlock\\_p](#) *rwlock*)

Definition at line 46 of file [mc\\_rwlock.c](#).

References [CHECK\\_NULL](#), [mc\\_rwlock\\_s::cond](#), [COND\\_INIT](#), [COND\\_T](#), [mc\\_rwlock\\_s::lock](#), [MC\\_ERR\\_MEMORY](#), [MUTEX\\_INIT](#), [MUTEX\\_T](#), [mc\\_rwlock\\_s::num\\_readers](#), [mc\\_rwlock\\_s::write\\_flag](#), and [mc\\_rwlock\\_s::write\\_request](#).

**13.42.2.3 int mc\_rwlock\_rdlock (mc\_rwlock\_p *rwlock*)**

Definition at line 79 of file mc\_rwlock.c.

References mc\_rwlock\_s::cond, COND\_WAIT, mc\_rwlock\_s::lock, MUTEX\_LOCK, MUTEX\_UNLOCK, mc\_rwlock\_s::num\_readers, mc\_rwlock\_s::write\_flag, and mc\_rwlock\_s::write\_request.

**13.42.2.4 int mc\_rwlock\_rdunlock (mc\_rwlock\_p *rwlock*)**

Definition at line 94 of file mc\_rwlock.c.

References mc\_rwlock\_s::cond, COND\_SIGNAL, mc\_rwlock\_s::lock, MUTEX\_LOCK, MUTEX\_UNLOCK, and mc\_rwlock\_s::num\_readers.

**13.42.2.5 int mc\_rwlock\_wrlock (mc\_rwlock\_p *rwlock*)**

Definition at line 107 of file mc\_rwlock.c.

References mc\_rwlock\_s::cond, COND\_WAIT, mc\_rwlock\_s::lock, MUTEX\_LOCK, MUTEX\_UNLOCK, mc\_rwlock\_s::num\_readers, mc\_rwlock\_s::write\_flag, and mc\_rwlock\_s::write\_request.

**13.42.2.6 int mc\_rwlock\_wrunlock (mc\_rwlock\_p *rwlock*)**

Definition at line 126 of file mc\_rwlock.c.

References mc\_rwlock\_s::cond, COND\_SIGNAL, mc\_rwlock\_s::lock, MUTEX\_LOCK, MUTEX\_UNLOCK, and mc\_rwlock\_s::write\_flag.

## 13.43 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/message.h File Reference

```
#include "config.h"
#include "mtp_http.h"
#include <mxml.h>
#include <netinet/in.h>
#include "security/interface.h"
```

### Data Structures

- struct [message\\_s](#)
- struct [message\\_send\\_arg\\_s](#)

### Typedefs

- typedef enum [message\\_type\\_e](#) [message\\_type\\_t](#)
- typedef struct [message\\_s](#) [message\\_t](#)
- typedef [message\\_t](#) \* [message\\_p](#)
- typedef struct [message\\_send\\_arg\\_s](#) [message\\_send\\_arg\\_t](#)

### Enumerations

- enum [message\\_type\\_e](#) {  
    [RELAY](#), [REQUEST](#), [SUBSCRIBE](#), [CANCEL](#),  
    [N\\_UNDRSTD](#), [MOBILE\\_AGENT](#), [QUER\\_IF](#), [QUER\\_REF](#),  
    [AGENT\\_UPDATE](#), [RETURN\\_MSG](#), [FIPA\\_ACL](#), [ENCRYPTED\\_DATA](#),  
    [ENCRYPTION\\_INITIALIZE](#), [REQUEST\\_ENCRYPTION\\_INITIALIZE](#), [NUM\\_MESSAGE\\_-](#)  
    [TYPE](#), [MSG\\_SEC](#) }

### Functions

- [message\\_p](#) [message\\_New](#) (void)
- [message\\_p](#) [message\\_Copy](#) ([message\\_p](#) src)
- int [message\\_InitializeFromAgent](#) (struct [mc\\_platform\\_s](#) \*[mc\\_platform](#), [message\\_p](#) message, struct [agent\\_s](#) \*[agent](#))
- int [message\\_InitializeFromConnection](#) (struct [mc\\_platform\\_s](#) \*[mc\\_platform](#), [message\\_p](#) message, struct [connection\\_s](#) \*[connection](#))
- int [message\\_InitializeFromString](#) (struct [mc\\_platform\\_s](#) \*[mc\\_platform](#), [message\\_p](#) message, const char \*string, const char \*destination\_host, int destination\_port, const char \*target)
- int [message\\_Destroy](#) ([message\\_p](#) message)
- int [auth\\_rece\\_send\\_msg](#) (int sockfd, char \*hostname, char \*message, char \*privkey, char \*known\_host\_filename)
- int [message\\_Send](#) (struct [mc\\_platform\\_s](#) \*[mc\\_platform](#), [message\\_p](#) message, char \*privatekey)
- void \* [message\\_send\\_Thread](#) (void \*arg)
- int [http\\_to\\_hostport](#) (const char \*http\_str, char \*\*host, int \*port, char \*\*target)

### 13.43.1 Typedef Documentation

#### 13.43.1.1 typedef message\_t\* message\_p

Definition at line 114 of file message.h.

#### 13.43.1.2 typedef struct message\_send\_arg\_s message\_send\_arg\_t

#### 13.43.1.3 typedef struct message\_s message\_t

#### 13.43.1.4 typedef enum message\_type\_e message\_type\_t

### 13.43.2 Enumeration Type Documentation

#### 13.43.2.1 enum message\_type\_e

Enumerator:

*RELAY*  
*REQUEST*  
*SUBSCRIBE*  
*CANCEL*  
*N\_UNDRSTD*  
*MOBILE\_AGENT*  
*QUER\_IF*  
*QUER\_REF*  
*AGENT\_UPDATE*  
*RETURN\_MSG*  
*FIPA\_ACL*  
*ENCRYPTED\_DATA*  
*ENCRYPTION\_INITIALIZE*  
*REQUEST\_ENCRYPTION\_INITIALIZE*  
*NUM\_MESSAGE\_TYPE*  
*MSG\_SEC*

Definition at line 51 of file message.h.

### 13.43.3 Function Documentation

#### 13.43.3.1 int auth\_rece\_send\_msg (int sockfd, char \* hostname, char \* message, char \* privkey, char \* known\_host\_filename)

Definition at line 437 of file message.c.

References `aes_en_de()`, `initiate_migration_process()`, `read_known_host_file()`, and `send_AES_en_MA()`.

Referenced by `message_send_Thread()`.

**13.43.3.2 int http\_to\_hostport (const char \* *http\_str*, char \*\* *host*, int \* *port*, char \*\* *target*)**

Definition at line 287 of file message.c.

References MC\_ERR\_PARSE.

Referenced by MC\_AclSend().

**13.43.3.3 message\_p message\_Copy (message\_p *src*)**

Definition at line 88 of file message.c.

**13.43.3.4 int message\_Destroy (message\_p *message*)**

Definition at line 398 of file message.c.

References message\_s::addr, message\_s::agent\_xml\_flag, message\_s::from\_address, MC\_SUCCESS, message\_s::message\_body, mxmlDelete(), message\_s::target, message\_s::to\_address, message\_s::update\_name, and message\_s::xml\_root.

Referenced by acc\_connection\_Thread(), acc\_MessageHandlerThread(), ams\_ManageAgentList(), MC\_LoadAgentFromFile(), MC\_SendAgentMigrationMessage(), MC\_SendAgentMigrationMessageFile(), message\_InitializeFromConnection(), and message\_InitializeFromString().

**13.43.3.5 int message\_InitializeFromAgent (struct mc\_platform\_s \* *mc\_platform*, message\_p *message*, struct agent\_s \* *agent*)**

Referenced by ams\_ManageAgentList().

**13.43.3.6 int message\_InitializeFromConnection (struct mc\_platform\_s \* *mc\_platform*, message\_p *message*, struct connection\_s \* *connection*)****13.43.3.7 int message\_InitializeFromString (struct mc\_platform\_s \* *mc\_platform*, message\_p *message*, const char \* *string*, const char \* *destination\_host*, int *destination\_port*, const char \* *target*)**

Referenced by MC\_LoadAgentFromFile(), MC\_SendAgentMigrationMessage(), and MC\_SendAgentMigrationMessageFile().

**13.43.3.8 message\_p message\_New (void)**

Definition at line 64 of file message.c.

References message\_s::addr, message\_s::agent\_xml\_flag, CHECK\_NULL, message\_s::connect\_id, message\_s::from\_address, message\_s::http\_type, message\_s::isHTTP, message\_s::message\_body, message\_s::message\_id, message\_s::message\_type, message\_s::target, message\_s::to\_address, message\_s::update\_name, message\_s::update\_num, message\_s::xml\_payload, and message\_s::xml\_root.

Referenced by acc\_connection\_Thread(), ams\_ManageAgentList(), MC\_LoadAgentFromFile(), MC\_SendAgentMigrationMessage(), MC\_SendAgentMigrationMessageFile(), and mtp\_http\_CreateMessage().

### 13.43.3.9 **int** message\_Send (struct mc\_platform\_s \* *mc\_platform*, message\_p *message*, char \* *privatekey*)

Definition at line 562 of file message.c.

References mc\_platform\_s::acc, COND\_WAIT, message\_send\_arg\_s::mc\_platform, message\_send\_arg\_s::message, message\_send\_Thread(), MSG\_THREADS, MUTEX\_LOCK, MUTEX\_UNLOCK, message\_send\_arg\_s::privatekey, THREAD\_CREATE, THREAD\_DETACH, and THREAD\_T.

Referenced by acc\_MessageHandlerThread(), MC\_AclSend(), and message\_queue\_SendOutgoing().

### 13.43.3.10 **void\*** message\_send\_Thread (void \* *arg*)

Definition at line 601 of file message.c.

References mc\_platform\_s::agency, auth\_rece\_send\_msg(), buf, CHECK\_NULL, dynstring\_Append(), dynstring\_Destroy(), dynstring\_New(), agency\_s::known\_host\_filename, mc\_platform, dynstring\_s::message, message\_s::message\_body, MSG\_THREAD\_EXIT, mtp\_http\_ComposeMessage(), mtp\_http\_Destroy(), mtp\_http\_New(), mtp\_http\_Parse(), port, send, message\_s::sending\_agent\_name, SOCKET\_ERROR, SOCKET\_INPUT\_SIZE, strtok\_r, and message\_s::to\_address.

Referenced by message\_Send().

## 13.44 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/mobilec.h File Reference

```
#include <dynstring.h>
#include <fipa_acl.h>
#include <fipa_comm.h>
```

## 13.45 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/mtp\_http.h File Reference

### Data Structures

- struct [mtp\\_http\\_content\\_s](#)
- struct [mtp\\_http\\_s](#)

### Defines

- #define [SOCKET\\_INPUT\\_SIZE](#) 4096

### Typedefs

- typedef struct [mtp\\_http\\_content\\_s](#) [mtp\\_http\\_content\\_t](#)
- typedef struct [mtp\\_http\\_s](#) [mtp\\_http\\_t](#)
- typedef [mtp\\_http\\_t](#) \* [mtp\\_http\\_p](#)

### Enumerations

- enum [http\\_status\\_code\\_e](#) {  
[CONTINUE](#) = 100, [SWITCHING\\_PROTOCOLS](#), [PROCESSING](#), [OK](#) = 200,  
[CREATED](#), [ACCEPTED](#), [NON\\_AUTHORITATIVE\\_INFORMATION](#), [NO\\_CONTENT](#),  
[RESET\\_CONTENT](#), [PARTIAL\\_CONTENT](#), [MULTI\\_STATUS](#), [BAD\\_REQUEST](#) = 400,  
[UNAUTHORIZED](#), [PAYMENT\\_REQUIRED](#), [FORBIDDEN](#), [NOT\\_FOUND](#),  
[METHOD\\_NOT\\_ALLOWED](#), [NOT\\_ACCEPTABLE](#), [PROXY\\_AUTHENTICATION\\_REQUIRED](#), [REQUEST\\_TIMEOUT](#),  
[CONFLICT](#), [GONE](#), [LENGTH\\_REQUIRED](#), [PRECONDITION\\_FAILED](#),  
[REQUEST\\_ENTITY\\_TOO\\_LARGE](#), [REQUEST\\_URI\\_TOO\\_LONG](#), [UNSUPPORTED\\_MEDIA\\_TYPE](#), [REQUESTED\\_RANGE\\_NOT\\_SATISFIABLE](#),  
[EXPECTATION\\_FAILED](#), [UNPROCESSABLE\\_ENTITY](#), [LOCKED](#), [FAILED\\_DEPENDANCY](#),  
[UNORDERED\\_COLLECTION](#), [UPGRADE\\_REQUIRED](#), [RETRY\\_WITH](#) }  
*http return status codes*

- enum [http\\_performative\\_e](#) {  
[HTTP\\_PERFORMATIVE\\_UNDEF](#) = -1, [HTTP\\_PERFORMATIVE\\_ZERO](#) = 0, [HTTP\\_HEAD](#),  
[HTTP\\_GET](#),  
[HTTP\\_POST](#), [HTTP\\_PUT](#), [HTTP\\_DELETE](#), [HTTP\\_TRACE](#),  
[HTTP\\_OPTIONS](#), [HTTP\\_CONNECT](#), [HTTP\\_RESPONSE](#), [HTTP\\_NUM\\_PERFORMATIVES](#) }  
*http 'verbs'*



## Functions

- const char \* [http\\_GetExpression](#) (const char \*string, char \*\*expr)  
*Parse an html expression.*
- int [http\\_ParseExpression](#) (const char \*expression\_string, char \*\*name, char \*\*value)  
*Parse an expression into its name and value.*
- const char \* [http\\_ParseRequest](#) (mtp\_http\_p http, const char \*string)
- const char \* [http\\_GetToken](#) (const char \*string, char \*\*token)
- int [mtp\\_http\\_Destroy](#) (mtp\_http\_p http)
- int [mtp\\_http\\_InitializeFromConnection](#) (struct [mtp\\_http\\_s](#) \*http, struct [connection\\_s](#) \*connection, char \*privatekey)
- [mtp\\_http\\_p](#) [mtp\\_http\\_New](#) (void)
- const char \* [mtp\\_http\\_ParseHeader](#) (struct [mtp\\_http\\_s](#) \*http, const char \*string)
- int [mtp\\_http\\_Parse](#) (struct [mtp\\_http\\_s](#) \*http, const char \*string)
- int [mtp\\_http\\_ComposeMessage](#) (struct [message\\_s](#) \*message)
- struct [message\\_s](#) \* [mtp\\_http\\_CreateMessage](#) (mtp\_http\_t \*mtp\_http, char \*hostname, int port)

### 13.45.1 Define Documentation

#### 13.45.1.1 #define SOCKET\_INPUT\_SIZE 4096

Definition at line 147 of file mtp\_http.h.

Referenced by [message\\_InitializeFromConnection\(\)](#), [message\\_send\\_Thread\(\)](#), and [mtp\\_http\\_InitializeFromConnection\(\)](#).

### 13.45.2 Typedef Documentation

#### 13.45.2.1 typedef struct mtp\_http\_content\_s mtp\_http\_content\_t

#### 13.45.2.2 typedef mtp\_http\_t\* mtp\_http\_p

Definition at line 145 of file mtp\_http.h.

#### 13.45.2.3 typedef struct mtp\_http\_s mtp\_http\_t

### 13.45.3 Enumeration Type Documentation

#### 13.45.3.1 enum http\_performative\_e

http 'verbs'

Enumerator:

*HTTP\_PERFORMATIVE\_UNDEF*  
*HTTP\_PERFORMATIVE\_ZERO*  
*HTTP\_HEAD*  
*HTTP\_GET*

*HTTP\_POST*  
*HTTP\_PUT*  
*HTTP\_DELETE*  
*HTTP\_TRACE*  
*HTTP\_OPTIONS*  
*HTTP\_CONNECT*  
*HTTP\_RESPONSE*  
*HTTP\_NUM\_PERFORMATIVES*

Definition at line 90 of file mtp\_http.h.

### 13.45.3.2 enum http\_status\_code\_e

http return status codes

**Enumerator:**

*CONTINUE*  
*SWITCHING\_PROTOCOLS*  
*PROCESSING*  
*OK*  
*CREATED*  
*ACCEPTED*  
*NON\_AUTHORITATIVE\_INFORMATION*  
*NO\_CONTENT*  
*RESET\_CONTENT*  
*PARTIAL\_CONTENT*  
*MULTI\_STATUS*  
*BAD\_REQUEST*  
*UNAUTHORIZED*  
*PAYMENT\_REQUIRED*  
*FORBIDDEN*  
*NOT\_FOUND*  
*METHOD\_NOT\_ALLOWED*  
*NOT\_ACCEPTABLE*  
*PROXY\_AUTHENTICATION\_REQUIRED*  
*REQUEST\_TIMEOUT*  
*CONFLICT*  
*GONE*  
*LENGTH\_REQUIRED*  
*PRECONDITION\_FAILED*  
*REQUEST\_ENTITY\_TOO\_LARGE*  
*REQUEST\_URI\_TOO\_LONG*

***UNSUPPORTED\_MEDIA\_TYPE***  
***REQUESTED\_RANGE\_NOT\_SATISFIABLE***  
***EXPECTATION\_FAILED***  
***UNPROCESSABLE\_ENTITY***  
***LOCKED***  
***FAILED\_DEPENDANCY***  
***UNORDERED\_COLLECTION***  
***UPGRADE\_REQUIRED***  
***RETRY\_WITH***

Definition at line 45 of file mtp\_http.h.

### 13.45.4 Function Documentation

#### 13.45.4.1 `const char* http_GetExpression (const char * string, char ** expr)`

Parse an html expression.

##### Parameters:

*string* (input) The html block of text: Will parse the first expression pointed to by 'string'.  
*expr* (output) The allocated expression

##### Returns:

A pointer to the next expression segment of the string block, or NULL.

Definition at line 347 of file mtp\_http.c.

Referenced by mtp\_http\_Parse(), and mtp\_http\_ParseHeader().

#### 13.45.4.2 `const char* http_GetToken (const char * string, char ** token)`

Definition at line 788 of file mtp\_http.c.

References cur.

Referenced by http\_ParseRequest().

#### 13.45.4.3 `int http_ParseExpression (const char * expression_string, char ** name, char ** value)`

Parse an expression into its name and value.

##### Parameters:

*expression\_string* (input) The expression  
*name* (output) An allocated name string or NULL  
*value* (output) An allocated value string or NULL

##### Returns:

error\_code\_t type

**Note:**

an http expression is something like 'Date: Mon, 23 May 2005 22:38:34 GMT'  
 ' where 'Date' is the name and the remainder of the string is the value

Definition at line 406 of file mtp\_http.c.

References CHECK\_NULL, MC\_ERR\_PARSE, and MC\_SUCCESS.

Referenced by mtp\_http\_Parse(), and mtp\_http\_ParseHeader().

**13.45.4.4 const char\* http\_ParseRequest (mtp\_http\_p *http*, const char \* *string*)**

Definition at line 699 of file mtp\_http.c.

References cur, HTTP\_CONNECT, HTTP\_DELETE, HTTP\_GET, http\_GetToken(), HTTP\_HEAD, HTTP\_OPTIONS, mtp\_http\_s::http\_performative, HTTP\_PERFORMATIVE\_UNDEF, HTTP\_POST, HTTP\_PUT, HTTP\_RESPONSE, HTTP\_TRACE, mtp\_http\_s::response\_code, mtp\_http\_s::response\_string, and mtp\_http\_s::target.

Referenced by mtp\_http\_ParseHeader().

**13.45.4.5 int mtp\_http\_ComposeMessage (struct message\_s \* *message*)**

Referenced by message\_send\_Thread().

**13.45.4.6 struct message\_s\* mtp\_http\_CreateMessage (mtp\_http\_t \* *mtp\_http*, char \* *hostname*, int *port*) [read]**

Definition at line 873 of file mtp\_http.c.

References buf, mtp\_http\_s::content, mtp\_http\_content\_s::content\_type, mtp\_http\_content\_s::data, dynstring\_Append(), dynstring\_Destroy(), dynstring\_New(), mtp\_http\_s::host, message\_s::isHTTP, dynstring\_s::len, dynstring\_s::message, message\_s::message\_body, message\_New(), mtp\_http\_s::message\_parts, PACKAGE\_VERSION, mtp\_http\_s::target, and message\_s::to\_address.

Referenced by MC\_AclSend().

**13.45.4.7 int mtp\_http\_Destroy (mtp\_http\_p *http*)**

Definition at line 54 of file mtp\_http.c.

References mtp\_http\_s::accept\_ranges, mtp\_http\_s::boundary, mtp\_http\_s::connection, mtp\_http\_s::content, mtp\_http\_s::content\_length, mtp\_http\_content\_s::content\_type, mtp\_http\_s::content\_type, mtp\_http\_content\_s::data, mtp\_http\_s::date, mtp\_http\_s::host, mtp\_http\_s::http\_version, mtp\_http\_s::message\_parts, mtp\_http\_s::response\_string, mtp\_http\_s::return\_code, SAFE\_FREE, mtp\_http\_s::server, mtp\_http\_s::target, and mtp\_http\_s::user\_agent.

Referenced by acc\_connection\_Thread(), MC\_AclSend(), message\_send\_Thread(), and mtp\_http\_InitializeFromConnection().

**13.45.4.8 int mtp\_http\_InitializeFromConnection (struct mtp\_http\_s \* *http*, struct connection\_s \* *connection*, char \* *privatekey*)**

Referenced by acc\_connection\_Thread().

#### **13.45.4.9 mtp\_http\_p mtp\_http\_New (void)**

Definition at line 87 of file mtp\_http.c.

References CHECK\_NULL, and mtp\_http\_s::content.

Referenced by acc\_connection\_Thread(), MC\_AclSend(), message\_send\_Thread(), and mtp\_http\_InitializeFromConnection().

#### **13.45.4.10 int mtp\_http\_Parse (struct mtp\_http\_s \* *http*, const char \* *string*)**

Definition at line 549 of file mtp\_http.c.

References mtp\_http\_s::boundary, mtp\_http\_s::content, mtp\_http\_s::content\_length, mtp\_http\_content\_s::content\_type, mtp\_http\_s::content\_type, mtp\_http\_content\_s::data, http\_GetExpression(), HTTP\_HEAD, http\_ParseExpression(), mtp\_http\_s::http\_performative, HTTP\_POST, HTTP\_PUT, HTTP\_RESPONSE, MC\_SUCCESS, mtp\_http\_s::message\_parts, and mtp\_http\_ParseHeader().

Referenced by message\_send\_Thread(), and mtp\_http\_InitializeFromConnection().

#### **13.45.4.11 const char\* mtp\_http\_ParseHeader (struct mtp\_http\_s \* *http*, const char \* *string*)**

Definition at line 465 of file mtp\_http.c.

References mtp\_http\_s::header\_length, http\_GetExpression(), HTTP\_PARSE\_EXPR, http\_ParseExpression(), http\_ParseRequest(), MC\_SUCCESS, and SAFE\_FREE.

Referenced by mtp\_http\_InitializeFromConnection(), and mtp\_http\_Parse().

## 13.46 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/xml\_compose.h File Reference

```
#include "agent.h"
```

### Functions

- [mxml\\_node\\_t \\* agent\\_xml\\_compose \(agent\\_p agent\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_gaf\\_message \(agent\\_p agent\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_message \(agent\\_p agent\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_mobile\\_agent \(agent\\_p agent\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_agent\\_data \(agent\\_p agent\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_name \(agent\\_p agent\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_owner \(agent\\_p agent\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_home \(agent\\_p agent\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_wg\\_code \(agent\\_p agent\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_tasks \(agent\\_p agent\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_task \(agent\\_p agent, int index\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_data \(agent\\_p agent, int index, interpreter\\_variable\\_data\\_t \\*interp\\_variable\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_agent\\_code \(agent\\_p agent, int index\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_row \(interpreter\\_variable\\_data\\_t \\*interp\\_variable, int index\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_create\\_row\\_nodes \(void \\*data, int index, int \\*extent, ChType\\_t type, int dim, int extent\\_index\)](#)

### 13.46.1 Function Documentation

#### 13.46.1.1 mxml\_node\_t\* agent\_xml\_compose (agent\_p agent)

Definition at line 46 of file `xml_compose.c`.

References `agent_xml_compose__gaf_message()`, `MXML_ADD_AFTER`, `MXML_ADD_TO_PARENT`, `MXML_NO_CALLBACK`, `mxmlAdd()`, `mxmlLoadString()`, and `node`.

Referenced by `message_InitializeFromAgent()`.

#### 13.46.1.2 mxml\_node\_t\* agent\_xml\_compose\_\_agent\_code (agent\_p agent, int index)

Definition at line 521 of file `xml_compose.c`.

References `agent_datastate_s::agent_code_ids`, `agent_datastate_s::agent_codes`, `agent_s::datastate`, `MXML_NO_PARENT`, `mxmlElementSetAttr()`, `mxmlNewElement()`, `node`, and `xml_new_cdata()`.

Referenced by `agent_xml_compose__tasks()`.

#### 13.46.1.3 mxml\_node\_t\* agent\_xml\_compose\_\_agent\_data (agent\_p agent)

Definition at line 150 of file `xml_compose.c`.

References `agent_xml_compose__home()`, `agent_xml_compose__name()`, `agent_xml_compose__owner()`, `agent_xml_compose__tasks()`, `agent_xml_compose__wg_code()`, `MXML_ADD_AFTER`, `mxmlAdd()`, `mxmlNewElement()`, and `node`.

Referenced by agent\_xml\_compose\_\_mobile\_agent().

#### 13.46.1.4 **mxml\_node\_t\* agent\_xml\_compose\_\_create\_row\_nodes** (void \* *data*, int *index*, int \* *extent*, ChType\_t *type*, int *dim*, int *extent\_index*)

Definition at line 572 of file xml\_compose.c.

References agent\_xml\_compose\_\_create\_row\_nodes(), buf, CH\_DATATYPE\_SIZE, CH\_DATATYPE\_VALUE\_STRING, MXML\_ADD\_AFTER, MXML\_ADD\_TO\_PARENT, MXML\_NO\_PARENT, mxm-  
lAdd(), mxmlElementSetAttr(), mxmlNewElement(), mxmlNewText(), node, and size.

Referenced by agent\_xml\_compose\_\_create\_row\_nodes(), and agent\_xml\_compose\_\_row().

#### 13.46.1.5 **mxml\_node\_t\* agent\_xml\_compose\_\_data** (agent\_p *agent*, int *index*, interpreter\_variable\_data\_t \* *interp\_variable*)

Definition at line 447 of file xml\_compose.c.

References agent\_xml\_compose\_\_row(), interpreter\_variable\_data\_s::array\_dim, buf, CH\_DATATYPE\_STRING, CH\_DATATYPE\_VALUE\_STRING, interpreter\_variable\_data\_s::data, interpreter\_variable\_data\_s::data\_type, MXML\_ADD\_AFTER, mxmlAdd(), mxmlElementSetAttr(), mxmlNewElement(), interpreter\_variable\_data\_s::name, and node.

Referenced by agent\_xml\_compose\_\_task().

#### 13.46.1.6 **mxml\_node\_t\* agent\_xml\_compose\_\_gaf\_message** (agent\_p *agent*)

Definition at line 66 of file xml\_compose.c.

References agent\_xml\_compose\_\_message(), MXML\_ADD\_AFTER, mxmlAdd(), mxmlNewElement(), and node.

Referenced by agent\_xml\_compose().

#### 13.46.1.7 **mxml\_node\_t\* agent\_xml\_compose\_\_home** (agent\_p *agent*)

Definition at line 253 of file xml\_compose.c.

References agent\_s::home, mxmlNewElement(), mxmlNewText(), and node.

Referenced by agent\_xml\_compose\_\_agent\_data().

#### 13.46.1.8 **mxml\_node\_t\* agent\_xml\_compose\_\_message** (agent\_p *agent*)

Definition at line 85 of file xml\_compose.c.

References agent\_s::agent\_type, agent\_xml\_compose\_\_mobile\_agent(), MC\_LOCAL\_AGENT, MC\_REMOTE\_AGENT, MC\_RETURN\_AGENT, MXML\_ADD\_AFTER, mxmlAdd(), mxmlElementSetAttr(), mxmlNewElement(), and node.

Referenced by agent\_xml\_compose\_\_gaf\_message().

#### 13.46.1.9 **mxml\_node\_t\* agent\_xml\_compose\_\_mobile\_agent** (agent\_p *agent*)

Definition at line 129 of file xml\_compose.c.

References `agent_xml_compose__agent_data()`, `MXML_ADD_AFTER`, `mxmlAdd()`, `mxmlNewElement()`, and `node`.

Referenced by `agent_xml_compose__message()`.

#### 13.46.1.10 `mxml_node_t* agent_xml_compose__name (agent_p agent)`

Definition at line 221 of file `xml_compose.c`.

References `mxmlNewElement()`, `mxmlNewText()`, `agent_s::name`, and `node`.

Referenced by `agent_xml_compose__agent_data()`.

#### 13.46.1.11 `mxml_node_t* agent_xml_compose__owner (agent_p agent)`

Definition at line 237 of file `xml_compose.c`.

References `mxmlNewElement()`, `mxmlNewText()`, `node`, and `agent_s::owner`.

Referenced by `agent_xml_compose__agent_data()`.

#### 13.46.1.12 `mxml_node_t* agent_xml_compose__row (interpreter_variable_data_t * interp_variable, int index)`

Definition at line 549 of file `xml_compose.c`.

References `agent_xml_compose__create_row_nodes()`, `interpreter_variable_data_s::array_dim`, `interpreter_variable_data_s::array_extent`, `interpreter_variable_data_s::data`, `interpreter_variable_data_s::data_type`, and `node`.

Referenced by `agent_xml_compose__data()`.

#### 13.46.1.13 `mxml_node_t* agent_xml_compose__task (agent_p agent, int index)`

Definition at line 342 of file `xml_compose.c`.

References `agent_task_s::agent_return_data`, `agent_task_s::agent_variable_list`, `agent_xml_compose__data()`, `buf`, `agent_task_s::code_id`, `agent_s::datastate`, `MXML_ADD_AFTER`, `mxmlAdd()`, `mxmlElementSetAttr()`, `mxmlNewElement()`, `node`, `agent_task_s::persistent`, `agent_datastate_s::persistent`, `agent_task_s::server_name`, `agent_datastate_s::tasks`, and `agent_task_s::var_name`.

Referenced by `agent_xml_compose__tasks()`.

#### 13.46.1.14 `mxml_node_t* agent_xml_compose__tasks (agent_p agent)`

Definition at line 285 of file `xml_compose.c`.

References `agent_xml_compose__agent_code()`, `agent_xml_compose__task()`, `buf`, `agent_s::datastate`, `MXML_ADD_AFTER`, `mxmlAdd()`, `mxmlElementSetAttr()`, `mxmlNewElement()`, `node`, `agent_datastate_s::number_of_tasks`, and `agent_datastate_s::task_progress`.

Referenced by `agent_xml_compose__agent_data()`.



**13.46.1.15** `mxml_node_t* agent_xml_compose__wg_code (agent_p agent)`

Definition at line 269 of file `xml_compose.c`.

References `mxmlNewElement()`, `mxmlNewText()`, `node`, and `agent_s::wg_code`.

Referenced by `agent_xml_compose__agent_data()`.

## 13.47 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/xml\_helper.h File Reference

```
#include <mxml.h>
```

### Functions

- [mxml\\_node\\_t \\* xml\\_find\\_sibling](#) (const [mxml\\_node\\_t](#) \**node*, const char \**sibling\_name*)
- char \* [xml\\_get\\_cdata](#) (const [mxml\\_node\\_t](#) \**node*)
- [mxml\\_node\\_t \\* xml\\_get\\_child](#) (const [mxml\\_node\\_t](#) \**node*, const char \**child\_name*, [int](#) *descend*)
- [mxml\\_node\\_t \\* xml\\_get\\_deep\\_child](#) (const [mxml\\_node\\_t](#) \**parent*, const char \*\**child\_path*)
- [mxml\\_node\\_t \\* xml\\_get\\_next\\_element](#) (const [mxml\\_node\\_t](#) \**node*)
- char \* [xml\\_get\\_text](#) (const [mxml\\_node\\_t](#) \**node*)
- const char \* [xml\\_get\\_element\\_name](#) (const [mxml\\_node\\_t](#) \**node*)
- [mxml\\_node\\_t \\* xml\\_new\\_cdata](#) ([mxml\\_node\\_t](#) \**parent*, const char \**text*)
- const char \* [whitespace\\_cb](#) ([mxml\\_node\\_t](#) \**node*, [int](#) *where*)

### 13.47.1 Function Documentation

#### 13.47.1.1 const char \* whitespace\_cb (mxml\_node\_t \* node, int where)

Definition at line 571 of file testmxml.c.

References [mxml\\_node\\_s::child](#), [mxml\\_value\\_u::element](#), [MXML\\_WS\\_AFTER\\_CLOSE](#), [MXML\\_WS\\_AFTER\\_OPEN](#), [MXML\\_WS\\_BEFORE\\_CLOSE](#), [MXML\\_WS\\_BEFORE\\_OPEN](#), [mxml\\_value\\_s::name](#), [mxml\\_node\\_s::parent](#), and [mxml\\_node\\_s::value](#).

Referenced by [main\(\)](#).

#### 13.47.1.2 mxml\_node\_t\* xml\_find\_sibling (const mxml\_node\_t \* node, const char \* sibling\_name)

Definition at line 54 of file xml\_helper.c.

References [MXML\\_NO\\_DESCEND](#), [mxmlFindElement\(\)](#), [mxml\\_node\\_s::parent](#), and [xml\\_get\\_element\\_name\(\)](#).

#### 13.47.1.3 char\* xml\_get\_cdata (const mxml\_node\_t \* node)

Definition at line 76 of file xml\_helper.c.

References [buf](#), [CHECK\\_NULL](#), [MXML\\_ELEMENT](#), [mxml\\_node\\_s::type](#), and [xml\\_get\\_element\\_name\(\)](#).

Referenced by [xml\\_get\\_text\(\)](#).

#### 13.47.1.4 mxml\_node\_t\* xml\_get\_child (const mxml\_node\_t \* node, const char \* child\_name, int descend)

Definition at line 109 of file xml\_helper.c.

References [mxmlFindElement\(\)](#).

Referenced by `agent_return_xml_parse()`, `agent_xml_parse__agent_data()`, `agent_xml_parse__data()`, `agent_xml_parse__mobile_agent()`, `message_xml_parse()`, `message_xml_parse__message()`, and `xml_get_deep_child()`.

#### **13.47.1.5** `mxml_node_t* xml_get_deep_child (const mxml_node_t * parent, const char ** child_path)`

Definition at line 128 of file `xml_helper.c`.

References `MXML_NO_DESCEND`, `node`, and `xml_get_child()`.

#### **13.47.1.6** `const char* xml_get_element_name (const mxml_node_t * node)`

Definition at line 222 of file `xml_helper.c`.

References `mxml_value_u::element`, `MXML_ELEMENT`, `mxml_value_s::name`, `mxml_node_s::type`, and `mxml_node_s::value`.

Referenced by `agent_xml_parse__data()`, `agent_xml_parse__mobile_agent()`, `agent_xml_parse__row()`, `message_xml_parse()`, `xml_find_sibling()`, and `xml_get_cdata()`.

#### **13.47.1.7** `mxml_node_t* xml_get_next_element (const mxml_node_t * node)`

Definition at line 142 of file `xml_helper.c`.

References `MXML_ELEMENT`, `mxml_node_s::next`, and `mxml_node_s::type`.

#### **13.47.1.8** `char* xml_get_text (const mxml_node_t * node)`

Definition at line 160 of file `xml_helper.c`.

References `CHECK_NULL`, `mxml_node_s::child`, `mxml_value_u::element`, `MXML_ELEMENT`, `MXML_TEXT`, `mxml_value_s::name`, `mxml_node_s::next`, `mxml_text_s::string`, `mxml_value_u::text`, `mxml_node_s::type`, `mxml_node_s::value`, and `xml_get_cdata()`.

Referenced by `agent_xml_parse__agent_code()`, `agent_xml_parse__home()`, `agent_xml_parse__name()`, `agent_xml_parse__owner()`, `agent_xml_parse__sender()`, and `agent_xml_parse__wg_code()`.

#### **13.47.1.9** `mxml_node_t* xml_new_cdata (mxml_node_t * parent, const char * text)`

Definition at line 235 of file `xml_helper.c`.

References `CHECK_NULL`, `mxmlNewElement()`, and `node`.

Referenced by `agent_xml_compose__agent_code()`.

## 13.48 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/include/xml\_parser.h File Reference

```
#include <mxml.h>
#include "macros.h"
#include "agent.h"
```

### Functions

- [STRUCT](#) (xml\_parser, const [mxml\\_node\\_t](#) \*root; const [mxml\\_node\\_t](#) \*node;)
- [error\\_code\\_t agent\\_xml\\_parse](#) ([agent\\_p](#) agent)
- [error\\_code\\_t agent\\_xml\\_parse\\_\\_mobile\\_agent](#) ([agent\\_p](#) agent, [xml\\_parser\\_p](#) xml\_parser)
- [error\\_code\\_t agent\\_xml\\_parse\\_\\_agent\\_data](#) ([agent\\_p](#) agent, [xml\\_parser\\_p](#) xml\_parser)
- [error\\_code\\_t agent\\_xml\\_parse\\_\\_name](#) ([agent\\_p](#) agent, [xml\\_parser\\_p](#) xml\_parser)
- [error\\_code\\_t agent\\_xml\\_parse\\_\\_owner](#) ([agent\\_p](#) agent, [xml\\_parser\\_p](#) xml\_parser)
- [error\\_code\\_t agent\\_xml\\_parse\\_\\_home](#) ([agent\\_p](#) agent, [xml\\_parser\\_p](#) xml\_parser)
- [error\\_code\\_t agent\\_xml\\_parse\\_\\_sender](#) ([agent\\_p](#) agent, [xml\\_parser\\_p](#) xml\_parser)
- [error\\_code\\_t agent\\_xml\\_parse\\_\\_wg\\_code](#) ([agent\\_p](#) agent, [xml\\_parser\\_p](#) xml\_parser)
- [error\\_code\\_t agent\\_xml\\_parse\\_\\_tasks](#) ([agent\\_p](#) agent, [xml\\_parser\\_p](#) xml\_parser)
- [error\\_code\\_t agent\\_xml\\_parse\\_\\_task](#) ([agent\\_p](#) agent, [xml\\_parser\\_p](#) xml\_parser, [int](#) index)
- [error\\_code\\_t agent\\_xml\\_parse\\_\\_data](#) ([agent\\_p](#) agent, [xml\\_parser\\_p](#) xml\_parser, [int](#) index)
- [error\\_code\\_t agent\\_xml\\_parse\\_\\_row](#) ([interpreter\\_variable\\_data\\_t](#) \*interp\_variable, [xml\\_parser\\_p](#) xml\_parser, [int](#) index)
- void [agent\\_xml\\_parse\\_\\_fill\\_row\\_data](#) (void \*data, [ChType\\_t](#) type, [int](#) \*extent, const [mxml\\_node\\_t](#) \*node, [int](#) \*index)
- [error\\_code\\_t agent\\_xml\\_parse\\_\\_agent\\_code](#) ([agent\\_p](#) agent, [int](#) index, [xml\\_parser\\_p](#) xml\_parser)
- [error\\_code\\_t message\\_xml\\_parse](#) ([message\\_p](#) message)
- [error\\_code\\_t message\\_xml\\_parse\\_\\_message](#) ([message\\_p](#) message, [xml\\_parser\\_p](#) xml\_parser)

### 13.48.1 Function Documentation

#### 13.48.1.1 [error\\_code\\_t agent\\_xml\\_parse](#) ([agent\\_p](#) agent)

Definition at line 52 of file [xml\\_parser.c](#).

References [agent\\_xml\\_parse\\_\\_mobile\\_agent\(\)](#), [agent\\_s::datastate](#), [MC\\_SUCCESS](#), and [agent\\_datastate\\_s::xml\\_agent\\_root](#).

Referenced by [agent\\_Initialize\(\)](#).

#### 13.48.1.2 [error\\_code\\_t agent\\_xml\\_parse\\_\\_agent\\_code](#) ([agent\\_p](#) agent, [int](#) index, [xml\\_parser\\_p](#) xml\_parser)

Definition at line 872 of file [xml\\_parser.c](#).

References [agent\\_datastate\\_s::agent\\_code](#), [agent\\_datastate\\_s::agent\\_code\\_ids](#), [agent\\_datastate\\_s::agent\\_codes](#), [agent\\_task\\_s::code\\_id](#), [agent\\_s::datastate](#), [MC\\_SUCCESS](#), [mxmlElementGetAttr\(\)](#), [agent\\_datastate\\_s::number\\_of\\_tasks](#), [agent\\_datastate\\_s::task\\_progress](#), [agent\\_datastate\\_s::tasks](#), and [xml\\_get\\_text\(\)](#).

Referenced by [agent\\_xml\\_parse\\_\\_tasks\(\)](#).

### 13.48.1.3 `error_code_t agent_xml_parse__agent_data (agent_p agent, xml_parser_p xml_parser)`

Definition at line 93 of file `xml_parser.c`.

References `agent_xml_parse__home()`, `agent_xml_parse__name()`, `agent_xml_parse__owner()`, `agent_xml_parse__sender()`, `agent_xml_parse__tasks()`, `agent_xml_parse__wg_code()`, `MC_ERR_PARSE`, `MC_SUCCESS`, and `xml_get_child()`.

Referenced by `agent_xml_parse__mobile_agent()`.

### 13.48.1.4 `error_code_t agent_xml_parse__data (agent_p agent, xml_parser_p xml_parser, int index)`

Definition at line 538 of file `xml_parser.c`.

References `agent_task_s::agent_return_data`, `agent_task_s::agent_variable_list`, `agent_xml_parse__row()`, `interpreter_variable_data_s::array_dim`, `CH_DATATYPE_SIZE`, `CH_DATATYPE_STR_TO_VAL`, `CH_STRING_DATATYPE`, `interpreter_variable_data_s::data`, `interpreter_variable_data_s::data_type`, `agent_s::datastate`, `interpreter_variable_data_New()`, `MC_ERR_PARSE`, `MC_SUCCESS`, `mxmlElementGetAttr()`, `interpreter_variable_data_s::name`, `mxml_node_s::parent`, `agent_task_s::persistent`, `agent_s::datastate_s::tasks`, `xml_get_child()`, and `xml_get_element_name()`.

Referenced by `agent_xml_parse__task()`.

### 13.48.1.5 `void agent_xml_parse__fill_row_data (void * data, ChType_t type, int * extent, const mxml_node_t * node, int * index)`

Definition at line 751 of file `xml_parser.c`.

References `agent_xml_parse__fill_row_data()`, `buf`, `CH_DATATYPE_SIZE`, `mxml_node_s::child`, `MXML_DESCEND_FIRST`, `MXML_ELEMENT`, `MXML_TEXT`, `mxmlFindElement()`, `mxml_text_s::string`, `strtok_r`, `mxml_value_u::text`, `mxml_node_s::type`, and `mxml_node_s::value`.

Referenced by `agent_xml_parse__fill_row_data()`, and `agent_xml_parse__row()`.

### 13.48.1.6 `error_code_t agent_xml_parse__home (agent_p agent, xml_parser_p xml_parser)`

Definition at line 219 of file `xml_parser.c`.

References `CHECK_NULL`, `agent_s::home`, `MC_SUCCESS`, and `xml_get_text()`.

Referenced by `agent_return_xml_parse()`, and `agent_xml_parse__agent_data()`.

### 13.48.1.7 `error_code_t agent_xml_parse__mobile_agent (agent_p agent, xml_parser_p xml_parser)`

Definition at line 65 of file `xml_parser.c`.

References `agent_xml_parse__agent_data()`, `MC_ERR_PARSE`, `xml_get_child()`, and `xml_get_element_name()`.

Referenced by `agent_xml_parse()`.

**13.48.1.8 error\_code\_t agent\_xml\_parse\_\_name (agent\_p agent, xml\_parser\_p xml\_parser)**

Definition at line 166 of file xml\_parser.c.

References CHECK\_NULL, MC\_ERR\_PARSE, MC\_SUCCESS, agent\_s::name, and xml\_get\_text().

Referenced by agent\_return\_xml\_parse(), and agent\_xml\_parse\_\_agent\_data().

**13.48.1.9 error\_code\_t agent\_xml\_parse\_\_owner (agent\_p agent, xml\_parser\_p xml\_parser)**

Definition at line 192 of file xml\_parser.c.

References CHECK\_NULL, MC\_SUCCESS, agent\_s::owner, and xml\_get\_text().

Referenced by agent\_return\_xml\_parse(), and agent\_xml\_parse\_\_agent\_data().

**13.48.1.10 error\_code\_t agent\_xml\_parse\_\_row (interpreter\_variable\_data\_t \* interp\_variable, xml\_parser\_p xml\_parser, int index)**

Definition at line 684 of file xml\_parser.c.

References agent\_xml\_parse\_\_fill\_row\_data(), interpreter\_variable\_data\_s::array\_dim, interpreter\_variable\_data\_s::array\_extent, CH\_DATATYPE\_SIZE, interpreter\_variable\_data\_s::data, interpreter\_variable\_data\_s::data\_type, MC\_SUCCESS, and xml\_get\_element\_name().

Referenced by agent\_xml\_parse\_\_data().

**13.48.1.11 error\_code\_t agent\_xml\_parse\_\_sender (agent\_p agent, xml\_parser\_p xml\_parser)**

Definition at line 245 of file xml\_parser.c.

References CHECK\_NULL, MC\_SUCCESS, agent\_s::sender, and xml\_get\_text().

Referenced by agent\_xml\_parse\_\_agent\_data().

**13.48.1.12 error\_code\_t agent\_xml\_parse\_\_task (agent\_p agent, xml\_parser\_p xml\_parser, int index)**

Definition at line 442 of file xml\_parser.c.

References agent\_xml\_parse\_\_data(), CHECK\_NULL, agent\_task\_s::code\_id, agent\_s::datastate, MC\_ERR\_PARSE, MC\_SUCCESS, MXML\_DESCEND\_FIRST, MXML\_NO\_DESCEND, mxmlElementGetAttr(), mxmlFindElement(), agent\_task\_s::persistent, agent\_task\_s::server\_name, agent\_datastate\_s::tasks, and agent\_task\_s::var\_name.

Referenced by agent\_xml\_parse\_\_tasks().

**13.48.1.13 error\_code\_t agent\_xml\_parse\_\_tasks (agent\_p agent, xml\_parser\_p xml\_parser)**

Definition at line 304 of file xml\_parser.c.

References agent\_datastate\_s::agent\_code, agent\_datastate\_s::agent\_code\_ids, agent\_datastate\_s::agent\_codes, agent\_task\_New(), agent\_xml\_parse\_\_agent\_code(), agent\_xml\_parse\_\_task(), buf, agent\_s::datastate, MC\_ERR\_PARSE, MXML\_DESCEND, MXML\_DESCEND\_FIRST, MXML\_NO\_DESCEND, mxmlElementGetAttr(), mxmlFindElement(), agent\_datastate\_s::number\_of\_tasks, agent\_datastate\_s::task\_progress, and agent\_datastate\_s::tasks.

Referenced by `agent_return_xml_parse()`, and `agent_xml_parse__agent_data()`.

**13.48.1.14 error\_code\_t agent\_xml\_parse\_\_wg\_code (agent\_p agent, xml\_parser\_p xml\_parser)**

Definition at line 273 of file `xml_parser.c`.

References `MC_SUCCESS`, `agent_s::wg_code`, and `xml_get_text()`.

Referenced by `agent_xml_parse__agent_data()`.

**13.48.1.15 error\_code\_t message\_xml\_parse (message\_p message)**

Definition at line 948 of file `xml_parser.c`.

References `MC_ERR_PARSE`, `message_xml_parse__message()`, `MXML_DESCEND`, `MXML_NO_DESCEND`, `mxmlFindElement()`, `xml_get_child()`, `xml_get_element_name()`, and `message_s::xml_root`.

Referenced by `acc_connection_Thread()`, `message_InitializeFromConnection()`, and `message_InitializeFromString()`.

**13.48.1.16 error\_code\_t message\_xml\_parse\_\_message (message\_p message, xml\_parser\_p xml\_parser)**

Definition at line 1003 of file `xml_parser.c`.

References `message_s::addr`, `buf`, `CHECK_NULL`, `ENCRYPTED_DATA`, `ENCRYPTION_INITIALIZE`, `FIPA_ACL`, `message_s::from_address`, `MC_ERR_PARSE`, `MC_SUCCESS`, `message_s::message_type`, `MOBILE_AGENT`, `mxmlElementGetAttr()`, `port`, `REQUEST_ENCRYPTION_INITIALIZE`, `RETURN_MSG`, `strtok_r`, `xml_get_child()`, and `message_s::xml_payload`.

Referenced by `message_xml_parse()`.

**13.48.1.17 STRUCT (xml\_parser, const mxml\_node\_t \*root;const mxml\_node\_t \*node;)**

## 13.49 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/libmc.c File Reference

```
#include "config.h"
#include <unistd.h>
#include <pthread.h>
#include <embedch.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <sys/time.h>
#include "include/libmc.h"
#include "include/macros.h"
#include "include/mc_platform.h"
#include "include/message.h"
#include "include/data_structures.h"
#include "include/fipa_acl_envelope.h"
#include "include/fipa_acl.h"
#include "include/agent.h"
#include "include/agent_task.h"
```

### Defines

- #define [HOST\\_NAME\\_MAX](#) 255

### Functions

- [int MC\\_AclDestroy](#) (struct [fipa\\_acl\\_message\\_s](#) \*message)  
*Destroy a FIPA ACL message.*
- EXPORTMC [fipa\\_acl\\_message\\_t](#) \* [MC\\_AclNew](#) (void)  
*Allocate a new ACL Message.*
- EXPORTMC [int MC\\_AclPost](#) ([MC\\_Agent\\_t](#) agent, struct [fipa\\_acl\\_message\\_s](#) \*message)  
*Post ACL message to agent.*
- EXPORTMC [fipa\\_acl\\_message\\_t](#) \* [MC\\_AclReply](#) ([fipa\\_acl\\_message\\_t](#) \*acl\_message)  
*Reply to an ACL message.*
- EXPORTMC [fipa\\_acl\\_message\\_t](#) \* [MC\\_AclRetrieve](#) ([MC\\_Agent\\_t](#) agent)  
*Retrieve an ACL message.*
- EXPORTMC [int MC\\_AclSend](#) ([MC\\_Agency\\_t](#) attr, [fipa\\_acl\\_message\\_t](#) \*acl)  
*Send a composed ACL Message.*



- EXPORTMC [fipa\\_acl\\_message\\_t](#) \* [MC\\_AclWaitRetrieve](#) ([MCAgent\\_t](#) agent)  
*Wait for and retrieve an ACL message.*
- [int](#) [MC\\_AclSetProtocol](#) ([fipa\\_acl\\_message\\_t](#) \*acl, enum [fipa\\_protocol\\_e](#) protocol)
- [int](#) [MC\\_AclSetConversationID](#) ([fipa\\_acl\\_message\\_t](#) \*acl, char \*id)
- [int](#) [MC\\_AclSetPerformative](#) ([fipa\\_acl\\_message\\_t](#) \*acl, enum [fipa\\_performative\\_e](#) performative)
- [int](#) [MC\\_AclSetSender](#) ([fipa\\_acl\\_message\\_t](#) \*acl, const char \*name, const char \*address)
- [int](#) [MC\\_AclAddReceiver](#) ([fipa\\_acl\\_message\\_t](#) \*acl, const char \*name, const char \*address)
- [int](#) [MC\\_AclAddReplyTo](#) ([fipa\\_acl\\_message\\_t](#) \*acl, const char \*name, const char \*address)
- [int](#) [MC\\_AclSetContent](#) ([fipa\\_acl\\_message\\_t](#) \*acl, const char \*content)
- EXPORTMC [int](#) [MC\\_AddAgent](#) ([MCAgency\\_t](#) attr, [MCAgent\\_t](#) agent)  
*Add an agent to the agency 'attr'.*
- [int](#) [MC\\_AddStationaryAgent](#) ([MCAgency\\_t](#) agency, void \*(\*agent\_thread)(struct [agent\\_thread\\_arg\\_s](#) \*), const char \*name, void \*agent\_args)
- const void \* [MC\\_AgentVariableRetrieve](#) ([MCAgent\\_t](#) agent, const char \*var\_name, [int](#) task\_num)  
*Retrieve a pointer to a previously saved variable.*
- [int](#) [MC\\_AgentVariableRetrieveInfo](#) ([MCAgent\\_t](#) agent, const char \*var\_name, [int](#) task\_num, const void \*\*data, [int](#) \*dim, const [int](#) \*\*extent)  
*Retrieve a info about a previously saved variable.*
- [int](#) [MC\\_AgentVariableSave](#) ([MCAgent\\_t](#) agent, const char \*var\_name)  
*Mark an agent variable for saving.*
- [int](#) [MC\\_Barrier](#) ([MCAgency\\_t](#) attr, [int](#) id)
- EXPORTMC [int](#) [MC\\_BarrierInit](#) ([MCAgency\\_t](#) attr, [int](#) id, [int](#) num\_procs)  
*Initialize a MobileC Barrier.*
- EXPORTMC [int](#) [MC\\_BarrierDelete](#) ([MCAgency\\_t](#) attr, [int](#) id)  
*Find and delete an initialized MobileC Barrier.*
- EXPORTMC [int](#) [MC\\_CallAgentFunc](#) ([MCAgent\\_t](#) agent, const char \*funcName, void \*returnVal, [int](#) numArgs,...)  
*Use custom ChOptions\_t type for internal Ch interpreter.*
- EXPORTMC [int](#) [MC\\_CallAgentFuncArg](#) ([MCAgent\\_t](#) agent, const char \*funcName, void \*returnVal, void \*arg)  
*Calls a function defined in an agent.*
- EXPORTMC [int](#) [MC\\_CallAgentFuncV](#) ([MCAgent\\_t](#) agent, const char \*funcName, void \*returnVal, va\_list ap)  
*Calls a function defined in an agent.*
- EXPORTMC [int](#) [MC\\_CallAgentFuncVar](#) ([MCAgent\\_t](#) agent, const char \*funcName, void \*returnVal, ChVaList\_t varg)
- EXPORTMC [MCAgent\\_t](#) [MC\\_ComposeAgent](#) (const char \*name, const char \*home, const char \*owner, const char \*code, const char \*return\_var\_name, const char \*server, [int](#) persistent)  
*Compose a new agent dynamically without using a prewritten XML file.*

- EXPORTMC [MCAgent\\_t MC\\_ComposeAgentS](#) (const char \*name, const char \*home, const char \*owner, const char \*code, const char \*return\_var\_name, const char \*server, const char \*workgroup\_code, [int](#) persistent)  
*Compose a new agent dynamically without using a prewritten XML file.*
- EXPORTMC [MCAgent\\_t MC\\_ComposeAgentFromFile](#) (const char \*name, const char \*home, const char \*owner, const char \*filename, const char \*return\_var\_name, const char \*server, [int](#) persistent)  
*Compose a new agent dynamically from a source code file.*
- EXPORTMC [MCAgent\\_t MC\\_ComposeAgentFromFileS](#) (const char \*name, const char \*home, const char \*owner, const char \*filename, const char \*return\_var\_name, const char \*server, const char \*workgroup\_code, [int](#) persistent)  
*Compose a new agent dynamically from a source code file.*
- EXPORTMC [int MC\\_CondBroadcast](#) ([MCAgency\\_t](#) attr, [int](#) id)  
*Wakes up all agents/threads waiting on a condition variable.*
- EXPORTMC [int MC\\_CondSignal](#) ([MCAgency\\_t](#) attr, [int](#) id)  
*Wakes up at least one thread waiting on a condition variable.*
- EXPORTMC [int MC\\_CondWait](#) ([MCAgency\\_t](#) attr, [int](#) id)  
*Wait on a MobileC synchronization variable.*
- EXPORTMC [int MC\\_CondReset](#) ([MCAgency\\_t](#) attr, [int](#) id)  
*Reset a previously signalled MobileC condition variable.*
- [int MC\\_CopyAgent](#) ([MCAgent\\_t](#) \*agent\_out, const [MCAgent\\_t](#) agent\_in)  
*Performs a deep-copy of an agent structure.*
- EXPORTMC [int MC\\_DeleteAgent](#) ([MCAgent\\_t](#) agent)  
*Stop and remove an agent.*
- EXPORTMC [int MC\\_DeleteAgentWG](#) ([MCAgent\\_t](#) calling\_agent, [MCAgent\\_t](#) agent)  
*Stop and remove an agent in the same workgroup.*
- [int MC\\_DestroyServiceSearchResult](#) (char \*\*agentName, char \*\*serviceName, [int](#) \*agentID, [int](#) numResult)  
*Free memory allocated by a Service Search operation.*
- [int MC\\_DeregisterService](#) ([MCAgency\\_t](#) agency, [int](#) agentID, const char \*serviceName)
- EXPORTMC [int MC\\_End](#) ([MCAgency\\_t](#) agency)  
*End an agency.*
- EXPORTMC [MCAgent\\_t MC\\_FindAgentByName](#) ([MCAgency\\_t](#) attr, const char \*name)  
*Find an agent by its name.*
- EXPORTMC [MCAgent\\_t MC\\_FindAgentByID](#) ([MCAgency\\_t](#) attr, [int](#) ID)  
*Find an agent by its id.*

- time\_t [MC\\_GetAgentArrivalTime](#) (MCAgent\_t agent)
- EXPORTMC int [MC\\_GetAgentStatus](#) (MCAgent\_t agent)  
*Get an agent's current status.*
- EXPORTMC char \* [MC\\_GetAgentXMLString](#) (MCAgent\_t agent)  
*Get an agent's xml string.*
- EXPORTMC void \* [MC\\_GetAgentExecEngine](#) (MCAgent\_t agent)  
*Retrieve an agent's Ch interpreter.*
- EXPORTMC int [MC\\_GetAgentID](#) (MCAgent\_t agent)  
*Retrieve an agent's id.*
- EXPORTMC char \* [MC\\_GetAgentName](#) (MCAgent\_t agent)
- EXPORTMC int [MC\\_GetAgentReturnData](#) (MCAgent\_t agent, int task\_num, void \*\*data, int \*dim, int \*\*extent)  
*Get an agent's return data.*
- EXPORTMC int [MC\\_GetAgentNumTasks](#) (MCAgent\_t agent)  
*Retrive the number of tasks an agent has.*
- EXPORTMC enum [MC\\_AgentType\\_e](#) [MC\\_GetAgentType](#) (MCAgent\_t agent)  
*Get an agent's type.*
- int [MC\\_GetAllAgents](#) (MCAgency\_t attr, MCAgent\_t \*\*agents, int \*num\_agents)
- EXPORTMC int [MC\\_HaltAgency](#) (MCAgency\_t attr)  
*Halt an agency: Do not process new entries in queues.*
- EXPORTMC MCAgency\_t [MC\\_Initialize](#) (int port, MCAgencyOptions\_t \*options)  
*Initialize and start a MobileC agency.*
- EXPORTMC int [MC\\_InitializeAgencyOptions](#) (struct MCAgencyOptions\_s \*options)  
*Initialize MobileC options.*
- EXPORTMC int [MC\\_LoadAgentFromFile](#) (MCAgency\_t attr, const char \*filename)  
*Load an agent from a file into an agency.*
- EXPORTMC int [MC\\_MigrateAgent](#) (MCAgent\_t agent, const char \*hostname, int port)  
*Migrates a running agent to another host.*
- EXPORTMC int [MC\\_MutexLock](#) (MCAgency\_t attr, int id)  
*Locks a MobileC synchronization variable as a mutex.*
- EXPORTMC int [MC\\_MutexUnlock](#) (MCAgency\_t attr, int id)
- EXPORTMC int [MC\\_PrintAgentCode](#) (MCAgent\_t agent)  
*Prints an agents code to stdout.*
- EXPORTMC int [MC\\_RegisterService](#) (MCAgency\_t agency, MCAgent\_t agent, int agentID, const char \*agentName, char \*\*serviceNames, int numServices)  
*Register a new service with the Directory Facilitator.*

- EXPORTMC [int MC\\_ResumeAgency](#) ([MCAgency\\_t](#) attr)  
*Resumes a halted agency.*
- EXPORTMC [MCAgency\\_t MC\\_RetrieveAgent](#) ([MCAgency\\_t](#) attr)  
*Retrieves the oldest agent from an agency.*
- EXPORTMC char \* [MC\\_RetrieveAgentCode](#) ([MCAgent\\_t](#) agent)  
*Retrieves an agent's Ch code.*
- EXPORTMC [int MC\\_ResetSignal](#) ([MCAgency\\_t](#) attr)  
*Reset a MobileC signal.*
- EXPORTMC [int MC\\_SearchForService](#) ([MCAgency\\_t](#) attr, const char \*searchString, char \*\*\*agentNames, char \*\*\*serviceNames, [int \\*\\*agentIDs](#), [int \\*numResults](#))  
*Search the directory facilitator for a service.*
- EXPORTMC [int MC\\_SemaphorePost](#) ([MCAgency\\_t](#) attr, [int](#) id)  
*Post to a MobileC synchronization variable semaphore.*
- EXPORTMC [int MC\\_SemaphoreWait](#) ([MCAgency\\_t](#) attr, [int](#) id)  
*Decreases a MobileC synchronization variable semaphore count by one.*
- [int MC\\_SendCh](#) ([MCAgency\\_t](#) attr, const char \*filename, const char \*remotehost, [int](#) port)
- EXPORTMC [int MC\\_SendAgentMigrationMessage](#) ([MCAgency\\_t](#) attr, const char \*string, const char \*hostname, [int](#) port)  
*Sends an agent migration message.*
- EXPORTMC [int MC\\_SendAgentMigrationMessageFile](#) ([MCAgency\\_t](#) attr, const char \*filename, const char \*hostname, [int](#) port)  
*Sends an agent migration message.*
- EXPORTMC [int MC\\_SendSteerCommand](#) ([MCAgency\\_t](#) attr, enum [MC\\_SteerCommand\\_e](#) cmd)
- [int MC\\_SetAgentStatus](#) ([MCAgent\\_t](#) agent, [int](#) status)  
*Set an agent's status.*
- [int MC\\_SetDefaultAgentStatus](#) ([MCAgency\\_t](#) agency, enum [MC\\_AgentStatus\\_e](#) status)  
*Sets default incoming agent status.*
- EXPORTMC [int MC\\_SetThreadOn](#) ([MCAgencyOptions\\_t](#) \*options, enum [MC\\_ThreadIndex\\_e](#) index)  
*Sets a MobileC thread to "on" status.*
- EXPORTMC [int MC\\_SetThreadsAllOn](#) ([MCAgencyOptions\\_t](#) \*options)  
*Set all Mobile-C threads on.*
- EXPORTMC [int MC\\_SetThreadOff](#) ([MCAgencyOptions\\_t](#) \*options, enum [MC\\_ThreadIndex\\_e](#) index)  
*Sets a MobileC thread to "off" status.*

- EXPORTMC [int MC\\_SetThreadsAllOff](#) ([MCAgencyOptions\\_t](#) \*options)  
*Set all MobileC threads to 'off' status.*
- EXPORTMC [int MC\\_Steer](#) ([MCAgency\\_t](#) attr, [int](#)(\*funcptr)(void \*data), void \*arg)  
*Set up a steerable algorithm.*
- EXPORTMC enum [MC\\_SteerCommand\\_e](#) [MC\\_SteerControl](#) (void)  
*The MobileC user-algorithm steering function.*
- EXPORTMC [int MC\\_SyncDelete](#) ([MCAgency\\_t](#) attr, [int](#) id)  
*Deletes a previously initialized synchronization variable.*
- EXPORTMC [int MC\\_SyncInit](#) ([MCAgency\\_t](#) attr, [int](#) id)  
*Initializes a new MobileC synchronization variable.*
- EXPORTMC [int MC\\_TerminateAgent](#) ([MCAgent\\_t](#) agent)  
*Halt a running agent.*
- EXPORTMC [int MC\\_TerminateAgentWG](#) ([MCAgent\\_t](#) calling\_agent, [MCAgent\\_t](#) agent)
- [int MC\\_MainLoop](#) ([MCAgency\\_t](#) attr)  
*Wait indefinitely.*
- EXPORTMC [int MC\\_WaitAgent](#) ([MCAgency\\_t](#) attr)  
*Wait indefinitely.*
- EXPORTMC [MCAgent\\_t](#) [MC\\_WaitRetrieveAgent](#) ([MCAgency\\_t](#) attr)  
*Wait and retrieve an agent.*
- EXPORTMC [int MC\\_WaitSignal](#) ([MCAgency\\_t](#) attr, [int](#) signals)  
*Wait for a MobileC signal.*
- [int MC\\_AclDestroy\\_chdl](#) (void \*varg)
- void \* [MC\\_AclNew\\_chdl](#) (void \*varg)
- [int MC\\_AclPost\\_chdl](#) (void \*varg)
- EXPORTCH void \* [MC\\_AclReply\\_chdl](#) (void \*varg)
- EXPORTCH void \* [MC\\_AclRetrieve\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_AclSend\\_chdl](#) (void \*varg)
- EXPORTCH void \* [MC\\_AclWaitRetrieve\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_AclSetProtocol\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_AclSetConversationID\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_AclSetPerformative\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_AclSetSender\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_AclAddReceiver\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_AclAddReplyTo\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_AclSetContent\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_AddAgent\\_chdl](#) (void \*varg)
- EXPORTCH const void \* [MC\\_AgentVariableRetrieve\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_AgentVariableSave\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_CallAgentFunc\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_Barrier\\_chdl](#) (void \*varg)

- EXPORTCH [int MC\\_BarrierDelete\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_BarrierInit\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_CondBroadcast\\_chdl](#) (void \*varg)
- EXPORTCH [MCAgent\\_t MC\\_ComposeAgent\\_chdl](#) (void \*varg)
- EXPORTCH [MCAgent\\_t MC\\_ComposeAgentS\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_CondSignal\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_CondReset\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_CondWait\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_DeleteAgent\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_DeleteAgentWG\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_DestroyServiceSearchResult\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_DeregisterService\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_End\\_chdl](#) (void \*varg)
- EXPORTCH [MCAgent\\_t MC\\_FindAgentByID\\_chdl](#) (void \*varg)
- EXPORTCH [MCAgent\\_t MC\\_FindAgentByName\\_chdl](#) (void \*varg)
- EXPORTCH [time\\_t MC\\_GetAgentArrivalTime\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_GetAgentID\\_chdl](#) (void \*varg)
- EXPORTCH [char \\* MC\\_GetAgentName\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_GetAgentNumTasks\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_GetAgentStatus\\_chdl](#) (void \*varg)
- EXPORTCH [char \\* MC\\_GetAgentXMLString\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_GetTimeOfDay\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_HaltAgency\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_MigrateAgent\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_MutexLock\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_MutexUnlock\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_PrintAgentCode\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_RegisterService\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_ResumeAgency\\_chdl](#) (void \*varg)
- EXPORTCH [MCAgent\\_t MC\\_RetrieveAgent\\_chdl](#) (void \*varg)
- EXPORTCH [char \\* MC\\_RetrieveAgentCode\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_SaveData\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_SearchForService\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_SemaphorePost\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_SemaphoreWait\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_SendAgentMigrationMessage\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_SendAgentMigrationMessageFile\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_SendSteerCommand\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_SetAgentStatus\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_SetDefaultAgentStatus\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_SyncDelete\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_SyncInit\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_TerminateAgent\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_TerminateAgentWG\\_chdl](#) (void \*varg)

## Variables

- [mc\\_platform\\_p g\\_mc\\_platform](#)

## 13.49.1 Define Documentation

### 13.49.1.1 `#define HOST_NAME_MAX 255`

Definition at line 67 of file libmc.c.

Referenced by MC\_Initialize().

## 13.49.2 Function Documentation

### 13.49.2.1 `int MC_AclAddReceiver (fipa_acl_message_t * acl, const char * name, const char * address)`

Definition at line 279 of file libmc.c.

References fipa\_agent\_identifier\_s::addresses, fipa\_agent\_identifier\_New(), fipa\_agent\_identifier\_set\_New(), fipa\_agent\_identifier\_set\_s::fipa\_agent\_identifiers, fipa\_url\_New(), fipa\_url\_sequence\_New(), fipa\_agent\_identifier\_s::name, fipa\_url\_sequence\_s::num, fipa\_agent\_identifier\_set\_s::num, fipa\_acl\_message\_s::receiver, fipa\_acl\_message\_s::receiver\_num, fipa\_url\_s::str, and fipa\_url\_sequence\_s::urls.

Referenced by MC\_AclAddReceiver\_chdl().

### 13.49.2.2 `EXPORTCH int MC_AclAddReceiver_chdl (void * varg)`

Definition at line 2269 of file libmc.c.

References MC\_AclAddReceiver().

Referenced by agent\_ChScriptInitVar().

### 13.49.2.3 `int MC_AclAddReplyTo (fipa_acl_message_t * acl, const char * name, const char * address)`

Definition at line 315 of file libmc.c.

References fipa\_agent\_identifier\_s::addresses, fipa\_agent\_identifier\_New(), fipa\_agent\_identifier\_set\_New(), fipa\_agent\_identifier\_set\_s::fipa\_agent\_identifiers, fipa\_url\_New(), fipa\_url\_sequence\_New(), fipa\_agent\_identifier\_s::name, fipa\_url\_sequence\_s::num, fipa\_agent\_identifier\_set\_s::num, fipa\_acl\_message\_s::reply\_to, fipa\_url\_s::str, and fipa\_url\_sequence\_s::urls.

Referenced by MC\_AclAddReplyTo\_chdl().

### 13.49.2.4 `EXPORTCH int MC_AclAddReplyTo_chdl (void * varg)`

Definition at line 2289 of file libmc.c.

References MC\_AclAddReplyTo().

Referenced by agent\_ChScriptInitVar().

### 13.49.2.5 `int MC_AclDestroy (struct fipa_acl_message_s * message)`

Destroy a FIPA ACL message.

**Parameters:**

*message* The ACL message to destroy

**Returns:**

0 on success, error code on failure.

Definition at line 78 of file libmc.c.

References fipa\_acl\_message\_Destroy().

Referenced by MC\_AclDestroy\_chdl().

**13.49.2.6 int MC\_AclDestroy\_chdl (void \* *varg*)**

Definition at line 2079 of file libmc.c.

References MC\_AclDestroy().

Referenced by agent\_ChScriptInitVar().

**13.49.2.7 EXPORTMC fipa\_acl\_message\_t\* MC\_AclNew (void) [read]**

Allocate a new ACL Message.

**Returns:**

A newly allocated and empty ACL message.

Definition at line 84 of file libmc.c.

References fipa\_acl\_message\_New().

Referenced by MC\_AclNew\_chdl().

**13.49.2.8 void\* MC\_AclNew\_chdl (void \* *varg*)**

Definition at line 2094 of file libmc.c.

References MC\_AclNew().

Referenced by agent\_ChScriptInitVar().

**13.49.2.9 EXPORTMC int MC\_AclPost (MCAgent\_t *agent*, struct fipa\_acl\_message\_s \* *message*)**

Post ACL message to agent.

**Parameters:**

*agent* The agent to post the message to

*message* The message to post

**Returns:**

0 if successful, or error\_code\_t type.



Definition at line 89 of file libmc.c.

References agent\_mailbox\_Post(), and agent\_s::mailbox.

Referenced by MC\_AclPost\_chdl(), and MC\_AclSend().

#### 13.49.2.10 int MC\_AclPost\_chdl (void \* *varg*)

Definition at line 2102 of file libmc.c.

References MC\_AclPost().

Referenced by agent\_ChScriptInitVar().

#### 13.49.2.11 EXPORTMC fipa\_acl\_message\_t\* MC\_AclReply (struct fipa\_acl\_message\_s \* *acl\_message*) [read]

Reply to an ACL message.

##### Parameters:

*acl\_message* The incoming acl message to reply to

##### Returns:

A newly allocated ACL message

##### Note:

This function simply generates a new ACL message with the 'receiver' field automatically set to the 'sender' field of the incoming message.

Definition at line 95 of file libmc.c.

References fipa\_Reply().

Referenced by MC\_AclReply\_chdl().

#### 13.49.2.12 EXPORTCH void\* MC\_AclReply\_chdl (void \* *varg*)

Definition at line 2120 of file libmc.c.

References MC\_AclReply().

Referenced by agent\_ChScriptInitVar().

#### 13.49.2.13 EXPORTMC fipa\_acl\_message\_t\* MC\_AclRetrieve (MCAgent\_t *agent*) [read]

Retrieve an ACL message.

##### Parameters:

*agent* Agent to retrieve message from.

##### Returns:

an ACL message struct on success or NULL on failure

Definition at line 101 of file libmc.c.

References agent\_mailbox\_Retrieve(), and agent\_s::mailbox.

Referenced by MC\_AclRetrieve\_chdl().

#### 13.49.2.14 EXPORTCH void\* MC\_AclRetrieve\_chdl (void \* *varg*)

Definition at line 2136 of file libmc.c.

References MC\_AclRetrieve().

Referenced by agent\_ChScriptInitVar().

#### 13.49.2.15 EXPORTMC int MC\_AclSend (MCAgency\_t *attr*, struct fipa\_acl\_message\_s \* *acl*)

Send a composed ACL Message.

##### Parameters:

*attr* An initialized and running MobileC agency

*acl* An allocated and fully composed ACL message.

##### Returns:

0 if successful, error code on failure.

Definition at line 107 of file libmc.c.

References fipa\_agent\_identifier\_s::addresses, mtp\_http\_s::content, mtp\_http\_content\_s::content\_type, mtp\_http\_content\_s::data, dynstring\_Destroy(), FIPA\_ACL, fipa\_acl\_Compose(), fipa\_acl\_message\_Copy(), fipa\_agent\_identifier\_set\_s::fipa\_agent\_identifiers, fipa\_envelope\_Compose(), mtp\_http\_s::host, http\_to\_hostport(), MC\_AclPost(), MC\_FindAgentByName(), agency\_s::mc\_platform, dynstring\_s::message, mtp\_http\_s::message\_parts, message\_Send(), message\_s::message\_type, mtp\_http\_CreateMessage(), mtp\_http\_Destroy(), mtp\_http\_New(), fipa\_agent\_identifier\_s::name, fipa\_url\_sequence\_s::num, fipa\_agent\_identifier\_set\_s::num, port, mc\_platform\_s::private\_key, fipa\_acl\_message\_s::receiver, fipa\_url\_s::str, mtp\_http\_s::target, message\_s::target, and fipa\_url\_sequence\_s::urls.

Referenced by MC\_AclSend\_chdl().

#### 13.49.2.16 EXPORTCH int MC\_AclSend\_chdl (void \* *varg*)

Definition at line 2152 of file libmc.c.

References CHECK\_NULL, MC\_AclSend(), and agency\_s::mc\_platform.

Referenced by agent\_ChScriptInitVar().

#### 13.49.2.17 int MC\_AclSetContent (fipa\_acl\_message\_t \* *acl*, const char \* *content*)

Definition at line 350 of file libmc.c.

References fipa\_string\_s::content, fipa\_acl\_message\_s::content, fipa\_string\_Destroy(), and fipa\_string\_New().

Referenced by MC\_AclSetContent\_chdl().

**13.49.2.18 EXPORTCH int MC\_AclSetContent\_chdl (void \* *varg*)**

Definition at line 2309 of file libmc.c.

References fipa\_acl\_message\_s::content, and MC\_AclSetContent().

Referenced by agent\_ChScriptInitVar().

**13.49.2.19 int MC\_AclSetConversationID (fipa\_acl\_message\_t \* *acl*, char \* *id*)**

Definition at line 229 of file libmc.c.

References fipa\_string\_s::content, fipa\_expression\_s::content, fipa\_acl\_message\_s::conversation\_id, FIPA\_EXPR\_STRING, fipa\_expression\_New(), fipa\_string\_New(), fipa\_expression\_s::content\_u::string, and fipa\_expression\_s::type.

Referenced by MC\_AclSetConversationID\_chdl().

**13.49.2.20 EXPORTCH int MC\_AclSetConversationID\_chdl (void \* *varg*)**

Definition at line 2213 of file libmc.c.

References MC\_AclSetConversationID().

Referenced by agent\_ChScriptInitVar().

**13.49.2.21 int MC\_AclSetPerformative (fipa\_acl\_message\_t \* *acl*, enum fipa\_performative\_e *performative*)**

Definition at line 248 of file libmc.c.

References fipa\_acl\_message\_s::performative.

Referenced by MC\_AclSetPerformative\_chdl().

**13.49.2.22 EXPORTCH int MC\_AclSetPerformative\_chdl (void \* *varg*)**

Definition at line 2231 of file libmc.c.

References MC\_AclSetPerformative(), and fipa\_acl\_message\_s::performative.

Referenced by agent\_ChScriptInitVar().

**13.49.2.23 int MC\_AclSetProtocol (fipa\_acl\_message\_t \* *acl*, enum fipa\_protocol\_e *protocol*)**

Definition at line 213 of file libmc.c.

References FIPA\_PROTOCOL\_END, FIPA\_PROTOCOL\_ERROR, and fipa\_acl\_message\_s::protocol.

Referenced by MC\_AclSetProtocol\_chdl().

**13.49.2.24 EXPORTCH int MC\_AclSetProtocol\_chdl (void \* *varg*)**

Definition at line 2194 of file libmc.c.

References MC\_AclSetProtocol(), and fipa\_acl\_message\_s::protocol.

Referenced by agent\_ChScriptInitVar().

### 13.49.2.25 int MC\_AclSetSender (fipa\_acl\_message\_t \* *acl*, const char \* *name*, const char \* *address*)

Definition at line 256 of file libmc.c.

References fipa\_agent\_identifier\_s::addresses, fipa\_agent\_identifier\_Destroy(), fipa\_agent\_identifier\_New(), fipa\_url\_New(), fipa\_url\_sequence\_New(), fipa\_agent\_identifier\_s::name, fipa\_url\_sequence\_s::num, fipa\_acl\_message\_s::sender, fipa\_url\_s::str, and fipa\_url\_sequence\_s::urls.

Referenced by MC\_AclSetSender\_chdl().

### 13.49.2.26 EXPORTCH int MC\_AclSetSender\_chdl (void \* *varg*)

Definition at line 2249 of file libmc.c.

References MC\_AclSetSender().

Referenced by agent\_ChScriptInitVar().

### 13.49.2.27 EXPORTMC fipa\_acl\_message\_t\* MC\_AclWaitRetrieve (MCAgent\_t *agent*) [read]

Wait for and retrieve an ACL message.

#### Parameters:

*agent* Agent to retrieve message from.

#### Returns:

an ACL message struct on success or NULL on failure

Definition at line 206 of file libmc.c.

References agent\_mailbox\_WaitRetrieve(), and agent\_s::mailbox.

Referenced by MC\_AclWaitRetrieve\_chdl().

### 13.49.2.28 EXPORTCH void\* MC\_AclWaitRetrieve\_chdl (void \* *varg*)

Definition at line 2176 of file libmc.c.

References MC\_AclWaitRetrieve().

Referenced by agent\_ChScriptInitVar().

### 13.49.2.29 EXPORTMC int MC\_AddAgent (MCAgency\_t *attr*, MCAgent\_t *agent*)

Add an agent to the agency 'attr'.

#### Parameters:

*attr* a MobileC agency

*agent* An initialized MobileC agent

**Returns:**

0 if successful, or `error_code_t` type

Definition at line 367 of file `libmc.c`.

References `mc_platform_s::agent_queue`, `mc_platform_s::ams`, `COND_SIGNAL`, `agency_s::mc_platform`, `agent_s::mc_platform`, `MUTEX_LOCK`, and `MUTEX_UNLOCK`.

Referenced by `MC_AddAgent_chdl()`.

**13.49.2.30 EXPORTCH int MC\_AddAgent\_chdl (void \* *varg*)**

Definition at line 2329 of file `libmc.c`.

References `CHECK_NULL`, `MC_AddAgent()`, and `agency_s::mc_platform`.

Referenced by `agent_ChScriptInitVar()`.

**13.49.2.31 int MC\_AddStationaryAgent (MCAgency\_t *agency*, void (\*)(struct agent\_thread\_arg\_s \*) *agent\_thread*, const char \* *name*, void \* *agent\_args*)**

Definition at line 381 of file `libmc.c`.

References `agent_thread_arg_s::agent`, `agent_NewBinary()`, `mc_platform_s::agent_queue`, `agent_thread_arg_s::args`, `agent_thread_arg_s::attr`, `agency_s::mc_platform`, `agent_s::name`, `agent_thread_arg_s::thread`, and `THREAD_CREATE`.

**13.49.2.32 const void\* MC\_AgentVariableRetrieve (MCAgent\_t *agent*, const char \* *var\_name*, int *task\_num*)**

Retrieve a pointer to a previously saved variable.

**Parameters:**

*agent* A MobileC agent.

*var\_name* The name of the saved variable that has previously been saved.

*task\_num* The previous completed task from which to retrieve the saved variable.

**Returns:**

A pointer to the data on success or `NULL` on failure.

**13.49.3 Examples**

The following example demonstrates usage of `MC_AgentVariableRetrieve()` from agent space.

Definition at line 402 of file `libmc.c`.

References `agent_task_s::agent_variable_list`, `interpreter_variable_data_s::data`, `agent_s::datastate`, `agent_datastate_s::task_progress`, and `agent_datastate_s::tasks`.

Referenced by `MC_AgentVariableRetrieve_chdl()`.

### 13.49.3.1 EXPORTCH const void\* MC\_AgentVariableRetrieve\_chdl (void \* *varg*)

Definition at line 2353 of file libmc.c.

References MC\_AgentVariableRetrieve().

Referenced by agent\_ChScriptInitVar().

### 13.49.3.2 int MC\_AgentVariableRetrieveInfo (MCAgent\_t *agent*, const char \* *var\_name*, int *task\_num*, const void \*\* *data*, int \* *dim*, const int \*\* *extent*)

Retrieve a info about a previously saved variable.

#### Parameters:

*agent* A MobileC agent.

*var\_name* The name of the saved variable that has previously been saved.

*task\_num* The previous completed task from which to retrieve the saved variable.

*data* (Output) The Variable Data

*dim* (Output) The dimension of the data array

*extent* (Output) The extents of the output array

#### Returns:

Error code.

Definition at line 420 of file libmc.c.

References agent\_task\_s::agent\_variable\_list, interpreter\_variable\_data\_s::array\_dim, interpreter\_variable\_data\_s::array\_extent, interpreter\_variable\_data\_s::data, agent\_s::datastate, MC\_ERR\_NOT\_FOUND, MC\_SUCCESS, agent\_datastate\_s::task\_progress, and agent\_datastate\_s::tasks.

### 13.49.3.3 int MC\_AgentVariableSave (MCAgent\_t *agent*, const char \* *var\_name*)

Mark an agent variable for saving.

#### Parameters:

*agent* A MobileC agent.

*var\_name* The name of the variable to mark for saving.

#### Returns:

0 on success, non-zero on failure.

#### See also:

test1.xml

## 13.49.4 Examples

See agent\_saved\_variables\_example/test1.xml for an example of usage of this api function.

### 13.49.5 Examples

The following example demonstrates usage of [MC\\_AgentVariableSave\(\)](#) from agent space.

Definition at line 441 of file libmc.c.

References `agent_s::datastate`, `MC_ERR_MEMORY`, `agent_task_s::num_saved_variables`, `agent_task_s::saved_variables`, `agent_datastate_s::task_progress`, and `agent_datastate_s::tasks`.

Referenced by `MC_AgentVariableSave_chdl()`.

#### 13.49.5.1 EXPORTCH int MC\_AgentVariableSave\_chdl (void \* *varg*)

Definition at line 2377 of file libmc.c.

References `MC_AgentVariableSave()`.

Referenced by `agent_ChScriptInitVar()`.

#### 13.49.5.2 int MC\_Barrier (MCAgency\_t *attr*, int *id*)

Definition at line 462 of file libmc.c.

References `mc_platform_s::barrier_queue`, `barrier_queue_Get()`, `barrier_node_s::cond`, `COND_BROADCAST`, `COND_WAIT`, `barrier_node_s::lock`, `MC_ERR_NOT_FOUND`, `agency_s::mc_platform`, `MC_SUCCESS`, `MUTEX_LOCK`, `MUTEX_UNLOCK`, `node`, `barrier_node_s::num_registered`, and `barrier_node_s::num_waiting`.

Referenced by `MC_Barrier_chdl()`.

#### 13.49.5.3 EXPORTCH int MC\_Barrier\_chdl (void \* *varg*)

Definition at line 2429 of file libmc.c.

References `CHECK_NULL`, `MC_Barrier()`, and `agency_s::mc_platform`.

Referenced by `agent_ChScriptInitVar()`.

#### 13.49.5.4 EXPORTMC int MC\_BarrierDelete (MCAgency\_t *attr*, int *id*)

Find and delete an initialized MobileC Barrier.

##### Parameters:

- attr* A running MobileC agency
- id* The id of the barrier node to delete

##### Returns:

returns 0 on success, error if the node is not found or other failure.

Definition at line 502 of file libmc.c.

References `mc_platform_s::barrier_queue`, `barrier_queue_Delete()`, and `agency_s::mc_platform`.

Referenced by `MC_BarrierDelete_chdl()`.

### 13.49.5.5 EXPORTCH int MC\_BarrierDelete\_chdl (void \* *varg*)

Definition at line 2452 of file libmc.c.

References CHECK\_NULL, MC\_BarrierDelete(), and agency\_s::mc\_platform.

Referenced by agent\_ChScriptInitVar().

### 13.49.5.6 EXPORTMC int MC\_BarrierInit (MCAgency\_t *attr*, int *id*, int *num\_procs*)

Initialize a MobileC Barrier.

#### Parameters:

*attr* A running MobileC agency

*id* The requested barrier id

*num\_procs* The number of agents/threads/processes that will wait on the barrier

#### Returns:

The allocated barrier id. May differ from the requested id if it is already in use.

## 13.49.6 Examples

The following example demonstrates an agent which sets up an MC\_Barrier.

Definition at line 488 of file libmc.c.

References barrier\_node\_Initialize(), mc\_platform\_s::barrier\_queue, barrier\_queue\_Add(), barrier\_queue\_Get(), MC\_ERR, agency\_s::mc\_platform, MC\_SUCCESS, and node.

Referenced by MC\_BarrierInit\_chdl().

### 13.49.6.1 EXPORTCH int MC\_BarrierInit\_chdl (void \* *varg*)

Definition at line 2475 of file libmc.c.

References CHECK\_NULL, MC\_BarrierInit(), and agency\_s::mc\_platform.

Referenced by agent\_ChScriptInitVar().

### 13.49.6.2 EXPORTMC int MC\_CallAgentFunc (MCAgent\_t *agent*, const char \* *funcName*, void \* *returnVal*, int *numArgs*, ...)

Use custom ChOptions\_t type for internal Ch interpreter.

#### Parameters:

*attr* A running MobileC agency

*options* Initialized Ch [options](#) structure

#### Returns:

0 on success, error\_code\_t type on failure Calls a function defined in an agent



**Parameters:**

*agent* An initialized and executed MobileC agent  
*funcName* The name of the function to call  
*returnVal* The agent function's return value  
*numArgs* The number of arguments supplied to the agent function  
... A variable argument list to be supplied to the agent function

**Returns:**

0 if successful, error\_code\_t type on failure

### 13.49.7 Example

Definition at line 508 of file libmc.c.

References agent\_s::agent\_interp, MUTEX\_LOCK, MUTEX\_UNLOCK, and agent\_s::run\_lock.

#### 13.49.7.1 EXPORTCH int MC\_CallAgentFunc\_chdl (void \* *varg*)

Definition at line 2399 of file libmc.c.

References MC\_CallAgentFuncVar().

Referenced by agent\_ChScriptInitVar().

#### 13.49.7.2 EXPORTMC int MC\_CallAgentFuncArg (MCAgent\_t *agent*, const char \* *funcName*, void \* *returnVal*, void \* *arg*)

Calls a function defined in an agent.

**Parameters:**

*agent* An initialized and executed MobileC agent  
*funcName* The name of the function to call  
*returnVal* The agent function's return value  
*arg* The agent functions argument

**Note:**

The agent function must be of the form 'void\* func(void\* arg);'

**Returns:**

0 if successful, error\_code\_t type on failure

### 13.49.8 Example

Definition at line 529 of file libmc.c.

References agent\_s::agent\_interp, MUTEX\_LOCK, MUTEX\_UNLOCK, and agent\_s::run\_lock.

### 13.49.8.1 EXPORTMC int MC\_CallAgentFuncV (MCAgent\_t *agent*, const char \* *funcName*, void \* *returnVal*, va\_list *ap*)

Calls a function defined in an agent.

#### Parameters:

*agent* An initialized and executed MobileC agent  
*funcName* The name of the function to call  
*returnVal* The agent function's return value  
*ap* A variable argument list

#### Returns:

0 if successful, error\_code\_t type on failure

Definition at line 548 of file libmc.c.

References agent\_s::agent\_interp, MUTEX\_LOCK, MUTEX\_UNLOCK, and agent\_s::run\_lock.

### 13.49.8.2 EXPORTMC int MC\_CallAgentFuncVar (MCAgent\_t *agent*, const char \* *funcName*, void \* *returnVal*, ChVaList\_t *varg*)

Definition at line 568 of file libmc.c.

References agent\_s::agent\_interp, MUTEX\_LOCK, MUTEX\_UNLOCK, and agent\_s::run\_lock.

Referenced by MC\_CallAgentFunc\_chdl().

### 13.49.8.3 EXPORTMC MCAgent\_t MC\_ComposeAgent (const char \* *name*, const char \* *home*, const char \* *owner*, const char \* *code*, const char \* *return\_var\_name*, const char \* *server*, int *persistent*)

Compose a new agent dynamically without using a prewritten XML file.

#### Parameters:

*name* The desired name of the new agent.  
*home* The home of the new agent.  
*owner* The owner of the new agent.  
*code* The agent code  
*return\_var\_name* The name of the agent's return variable. Set to "no-return" if no return variable is desired.  
*server* The target destination of the agent.  
*persistent* A flag indicating whether or not the agent should be persistent. A value of '1' indicates persistence, while a value of '0' indicates default non-persistent behaviour.

#### Returns:

This function returns a valid MCAgent\_t structure on success or NULL on failure.

Definition at line 608 of file libmc.c.

References MC\_ComposeAgentS().

#### 13.49.8.4 EXPORTCH MCAgent\_t MC\_ComposeAgent\_chdl (void \* *varg*)

Definition at line 2523 of file libmc.c.

References MC\_ComposeAgentS().

Referenced by agent\_ChScriptInitVar().

#### 13.49.8.5 EXPORTMC MCAgent\_t MC\_ComposeAgentFromFile (const char \* *name*, const char \* *home*, const char \* *owner*, const char \* *filename*, const char \* *return\_var\_name*, const char \* *server*, int *persistent*)

Compose a new agent dynamically from a source code file.

##### Parameters:

*filename* The filename of the source file

*name* The desired name of the new agent.

*home* The home of the new agent.

*owner* The owner of the new agent.

*code* The agent code

*return\_var\_name* The name of the agent's return variable. Set to "no-return" if no return variable is desired.

*server* The target destination of the agent.

*persistant* A flag indicating whether or not the agent should be persistent. A value of '1' indicates persistence, while a value of '0' indicates default non-persistent behaviour.

##### Returns:

This function returns a valid MCAgent\_t structure on success or NULL on failure.

Definition at line 707 of file libmc.c.

References MC\_ComposeAgentFromFileS().

#### 13.49.8.6 EXPORTMC MCAgent\_t MC\_ComposeAgentFromFileS (const char \* *name*, const char \* *home*, const char \* *owner*, const char \* *filename*, const char \* *return\_var\_name*, const char \* *server*, const char \* *workgroup\_code*, int *persistent*)

Compose a new agent dynamically from a source code file.

##### Parameters:

*filename* The filename of the source file

*name* The desired name of the new agent.

*home* The home of the new agent.

*owner* The owner of the new agent.

*code* The agent code

*return\_var\_name* The name of the agent's return variable. Set to "no-return" if no return variable is desired.

*server* The target destination of the agent.

***workgroup\_code*** The secret workgroup code of the agent. Only agents with the same workgroup code may perform certain interactions.

***persistant*** A flag indicating whether or not the agent should be persistent. A value of '1' indicates persistence, while a value of '0' indicates default non-persistent behaviour.

#### Returns:

This function returns a valid MCAgent\_t structure on success or NULL on failure.

Definition at line 728 of file libmc.c.

References agent\_thread\_arg\_s::agent, and MC\_ComposeAgentS().

Referenced by MC\_ComposeAgentFromFile().

**13.49.8.7 EXPORTMC MCAgent\_t MC\_ComposeAgentS (const char \* *name*, const char \* *home*, const char \* *owner*, const char \* *code*, const char \* *return\_var\_name*, const char \* *server*, const char \* *workgroup\_code*, int *persistant*)**

Compose a new agent dynamically without using a prewritten XML file.

#### Parameters:

***name*** The desired name of the new agent.

***home*** The home of the new agent.

***owner*** The owner of the new agent.

***code*** The agent code

***return\_var\_name*** The name of the agent's return variable. Set to "no-return" if no return variable is desired.

***server*** The target destination of the agent.

***workgroup\_code*** The secret workgroup code of the agent. Only agents with the same workgroup code may perform certain interactions.

***persistant*** A flag indicating whether or not the agent should be persistent. A value of '1' indicates persistence, while a value of '0' indicates default non-persistent behaviour.

#### Returns:

This function returns a valid MCAgent\_t structure on success or NULL on failure.

Definition at line 630 of file libmc.c.

References agent\_thread\_arg\_s::agent, agent\_datastate\_s::agent\_code, agent\_datastate\_s::agent\_code\_ids, agent\_datastate\_s::agent\_codes, agent\_datastate\_New(), agent\_New(), agent\_s::agent\_status, agent\_task\_New(), agent\_s::agent\_type, agent\_s::datastate, agent\_s::home, MC\_LOCAL\_AGENT, MC\_WAIT\_MESSGSEND, agent\_s::name, agent\_datastate\_s::number\_of\_tasks, agent\_s::orphan, agent\_s::owner, agent\_datastate\_s::persistent, agent\_task\_s::server\_name, agent\_datastate\_s::tasks, agent\_task\_s::var\_name, and agent\_s::wg\_code.

Referenced by MC\_ComposeAgent(), MC\_ComposeAgent\_chdl(), MC\_ComposeAgentFromFileS(), and MC\_ComposeAgentS\_chdl().

**13.49.8.8 EXPORTCH MC\_Agent\_t MC\_ComposeAgentS\_chdl (void \* *varg*)**

Definition at line 2564 of file libmc.c.

References `MC_ComposeAgentS()`.

Referenced by `agent_ChScriptInitVar()`.

**13.49.8.9 EXPORTMC int MC\_CondBroadcast (MC\_Agency\_t *attr*, int *id*)**

Wakes up all agents/threads waiting on a condition variable.

**Parameters:**

*attr* A MobileC agency

*id* Synchronization variable id to broadcast to

**See also:**

[MC\\_SyncInit\(\)](#), [MC\\_CondSignal\(\)](#)

**Returns:**

0 on success, `error_code_t` type on failure

Definition at line 767 of file libmc.c.

References `syncListNode_s::cond`, `COND_BROADCAST`, `syncListNode_s::lock`, `MC_ERR_NOT_FOUND`, `agency_s::mc_platform`, `MUTEX_LOCK`, `MUTEX_UNLOCK`, `syncListNode_s::signalled`, `mc_platform_s::syncList`, and `syncListFind()`.

Referenced by `MC_CondBroadcast_chdl()`.

**13.49.8.10 EXPORTCH int MC\_CondBroadcast\_chdl (void \* *varg*)**

Definition at line 2500 of file libmc.c.

References `CHECK_NULL`, `MC_CondBroadcast()`, and `agency_s::mc_platform`.

Referenced by `agent_ChScriptInitVar()`.

**13.49.8.11 EXPORTMC int MC\_CondReset (MC\_Agency\_t *attr*, int *id*)**

Reset a previously signalled MobileC condition variable.

**Parameters:**

*attr* A MobileC Agency

*id* The synchronization variable id to reset

**See also:**

[MC\\_SyncInit\(\)](#)

**Returns:**

0 on success, `error_code_t` type on failure

Definition at line 819 of file libmc.c.

References `syncListNode_s::lock`, `MC_ERR_NOT_FOUND`, `agency_s::mc_platform`, `MUTEX_LOCK`, `MUTEX_UNLOCK`, `syncListNode_s::signalled`, `mc_platform_s::syncList`, and `syncListFind()`.

Referenced by `MC_CondReset_chdl()`.

#### 13.49.8.12 EXPORTCH int MC\_CondReset\_chdl (void \* *varg*)

Definition at line 2630 of file libmc.c.

References `CHECK_NULL`, `MC_CondReset()`, and `agency_s::mc_platform`.

Referenced by `agent_ChScriptInitVar()`.

#### 13.49.8.13 EXPORTMC int MC\_CondSignal (MCAgency\_t *attr*, int *id*)

Wakes up at least one thread waiting on a condition variable.

##### Parameters:

*attr* A MobileC agency

*id* synchronization variable id

##### See also:

[MC\\_SyncInit\(\)](#)

##### Returns:

0 on success, `error_code_t` type on failure

### 13.49.9 Example

The following example demonstrates the agent-space version of the function, which is nearly identical to the binary space api function.

Definition at line 782 of file libmc.c.

References `syncListNode_s::cond`, `COND_SIGNAL`, `syncListNode_s::lock`, `MC_ERR_NOT_FOUND`, `agency_s::mc_platform`, `MUTEX_LOCK`, `MUTEX_UNLOCK`, `syncListNode_s::signalled`, `mc_platform_s::syncList`, and `syncListFind()`.

Referenced by `MC_CondSignal_chdl()`.

#### 13.49.9.1 EXPORTCH int MC\_CondSignal\_chdl (void \* *varg*)

Definition at line 2607 of file libmc.c.

References `CHECK_NULL`, `MC_CondSignal()`, and `agency_s::mc_platform`.

Referenced by `agent_ChScriptInitVar()`.

### 13.49.9.2 EXPORTMC int MC\_CondWait (MCAgency\_t *attr*, int *id*)

Wait on a MobileC synchronization variable.

#### Parameters:

*attr* A MobileC agency  
*id* a synchronization variable id

#### See also:

[MC\\_SyncInit\(\)](#)

#### Returns:

0 on success, error\_code\_t type on failure

## 13.49.10 Example

The following example demonstrates the agent-space version of this function.

Definition at line 797 of file libmc.c.

References syncListNode\_s::cond, COND\_WAIT, syncListNode\_s::lock, MC\_ERR\_NOT\_FOUND, agency\_s::mc\_platform, MUTEX\_LOCK, MUTEX\_UNLOCK, syncListNode\_s::signalled, mc\_platform\_s::syncList, and syncListFind().

Referenced by MC\_CondWait\_chdl().

### 13.49.10.1 EXPORTCH int MC\_CondWait\_chdl (void \* *varg*)

Definition at line 2653 of file libmc.c.

References CHECK\_NULL, MC\_CondWait(), and agency\_s::mc\_platform.

Referenced by agent\_ChScriptInitVar().

### 13.49.10.2 int MC\_CopyAgent (MCAgent\_t \* *agent\_out*, const MCAgent\_t *agent\_in*)

Performs a deep-copy of an agent structure.

#### Parameters:

*agent\_out* A pointer to the agent to copy to.  
*agent\_in* The agent to copy

#### Returns:

0 on success, error\_code\_t type on failure.

Definition at line 837 of file libmc.c.

References agent\_Copy(), and MC\_SUCCESS.

### 13.49.10.3 EXPORTMC int MC\_DeleteAgent (MCAgent\_t *agent*)

Stop and remove an agent.

#### Parameters:

*agent* An agent in any state (running, waiting, etc)

#### Returns:

0 on success, error\_code\_t type on failure

Definition at line 844 of file libmc.c.

References CHECK\_NULL, MC\_ERR\_INVALID, MC\_ERR\_INVALID\_ARGS, MC\_SetAgentStatus(), MC\_SUCCESS, MC\_TerminateAgent(), MC\_WAIT\_FINISHED, and agent\_s::wg\_code.

Referenced by MC\_DeleteAgent\_chdl().

### 13.49.10.4 EXPORTCH int MC\_DeleteAgent\_chdl (void \* *varg*)

Definition at line 2675 of file libmc.c.

References MC\_DeleteAgent(), MC\_ERR\_NOT\_FOUND, and MC\_FindAgentByName().

Referenced by agent\_ChScriptInitVar().

### 13.49.10.5 EXPORTMC int MC\_DeleteAgentWG (MCAgent\_t *calling\_agent*, MCAgent\_t *agent*)

Stop and remove an agent in the same workgroup.

#### Parameters:

*calling\_agent* The calling agent

*agent* An agent in any state (running, waiting, etc)

#### Note:

The agents must belong to the same workgroup.

#### Returns:

0 on success, error\_code\_t type on failure

Definition at line 864 of file libmc.c.

References CHECK\_NULL, MC\_ERR\_INVALID, MC\_ERR\_INVALID\_ARGS, MC\_SetAgentStatus(), MC\_SUCCESS, MC\_TerminateAgentWG(), MC\_WAIT\_FINISHED, and agent\_s::wg\_code.

Referenced by MC\_DeleteAgentWG\_chdl().

### 13.49.10.6 EXPORTCH int MC\_DeleteAgentWG\_chdl (void \* *varg*)

Definition at line 2695 of file libmc.c.

References MC\_DeleteAgentWG(), MC\_ERR\_NOT\_FOUND, and MC\_FindAgentByName().

Referenced by agent\_ChScriptInitVar().



**13.49.10.7 int MC\_DeregisterService (MCAgency\_t *agency*, int *agentID*, const char \* *serviceName*)**

Definition at line 907 of file libmc.c.

References mc\_platform\_s::df, df\_AddRequest(), df\_request\_list\_node\_New(), and agency\_s::mc\_platform.

Referenced by MC\_DeregisterService\_chdl().

**13.49.10.8 EXPORTCH int MC\_DeregisterService\_chdl (void \* *varg*)**

Definition at line 2744 of file libmc.c.

References CHECK\_NULL, MC\_DeregisterService(), and agency\_s::mc\_platform.

Referenced by agent\_ChScriptInitVar().

**13.49.10.9 int MC\_DestroyServiceSearchResult (char \*\* *agentName*, char \*\* *serviceName*, int \* *agentID*, int *numResult*)**

Free memory allocated by a Service Search operation.

**Parameters:**

*agentName* agent names returned by a search operation.

*serviceName* service names return by a search operation.

*agentID* list of agent id's returned by a search operation.

*numResult* The number of hits returned by a search operation.

**Returns:**

0 on success, error code on failure.

Definition at line 887 of file libmc.c.

Referenced by MC\_DestroyServiceSearchResult\_chdl().

**13.49.10.10 EXPORTCH int MC\_DestroyServiceSearchResult\_chdl (void \* *varg*)**

Definition at line 2717 of file libmc.c.

References MC\_DestroyServiceSearchResult().

Referenced by agent\_ChScriptInitVar().

**13.49.10.11 EXPORTMC int MC\_End (MCAgency\_t *attr*)**

End an agency.

**Parameters:**

*attr* A running agency

**Returns:**

0 on success, error\_code\_t type on failure

### 13.49.11 Example

Definition at line 936 of file libmc.c.

References `mc_platform_s::acc`, `mc_platform_s::ams`, `mc_platform_s::cmd_prompt`, `COND_BROADCAST`, `COND_SIGNAL`, `mc_platform_s::connection_queue`, `mc_platform_s::df`, `GET_THREAD_MODE`, `agency_s::hostName`, `agency_s::mc_platform`, `mc_platform_Destroy()`, `MC_THREAD_ACC`, `MC_THREAD_AMS`, `MC_THREAD_CP`, `MC_THREAD_DF`, `mc_platform_s::message_queue`, `MUTEX_LOCK`, `MUTEX_UNLOCK`, `mc_platform_s::quit`, `mc_platform_s::quit_cond`, `mc_platform_s::quit_lock`, `cmd_prompt_s::thread`, `THREAD_CANCEL`, `THREAD_JOIN`, and `agency_s::threads`.

Referenced by `MC_End_chdl()`.

#### 13.49.11.1 EXPORTCH int MC\_End\_chdl (void \* *varg*)

Definition at line 2771 of file libmc.c.

References `CHECK_NULL`, `MC_End()`, and `agency_s::mc_platform`.

Referenced by `agent_ChScriptInitVar()`.

#### 13.49.11.2 EXPORTMC MCAgent\_t MC\_FindAgentByID (MCAgency\_t *attr*, int *ID*)

Find an agent by its id.

##### Parameters:

*attr* the agency to search

*ID* the id to search for

##### Returns:

a valid agent on success, NULL on failure

Definition at line 1012 of file libmc.c.

References `mc_platform_s::agent_queue`, and `agency_s::mc_platform`.

Referenced by `MC_FindAgentByID_chdl()`.

#### 13.49.11.3 EXPORTCH MCAgent\_t MC\_FindAgentByID\_chdl (void \* *varg*)

Definition at line 2788 of file libmc.c.

References `CHECK_NULL`, `MC_FindAgentByID()`, and `agency_s::mc_platform`.

Referenced by `agent_ChScriptInitVar()`.

#### 13.49.11.4 EXPORTMC MCAgent\_t MC\_FindAgentByName (MCAgency\_t *attr*, const char \* *name*)

Find an agent by its name.

**Parameters:**

*attr* a running agency  
*name* name to search for

**Returns:**

a valid agent on success or NULL on failure

**13.49.12 Example**

Definition at line 999 of file libmc.c.

References mc\_platform\_s::agent\_queue, and agency\_s::mc\_platform.

Referenced by MC\_AclSend(), MC\_DeleteAgent\_chdl(), MC\_DeleteAgentWG\_chdl(), MC\_FindAgentByName\_chdl(), MC\_TerminateAgent\_chdl(), and MC\_TerminateAgentWG\_chdl().

**13.49.12.1 EXPORTCH MCAgent\_t MC\_FindAgentByName\_chdl (void \* *varg*)**

Definition at line 2811 of file libmc.c.

References CHECK\_NULL, MC\_FindAgentByName(), and agency\_s::mc\_platform.

Referenced by agent\_ChScriptInitVar().

**13.49.12.2 time\_t MC\_GetAgentArrivalTime (MCAgent\_t *agent*)**

Definition at line 1029 of file libmc.c.

References agent\_s::arrival\_time.

Referenced by MC\_GetAgentArrivalTime\_chdl().

**13.49.12.3 EXPORTCH time\_t MC\_GetAgentArrivalTime\_chdl (void \* *varg*)**

Definition at line 2838 of file libmc.c.

References MC\_GetAgentArrivalTime().

**13.49.12.4 EXPORTMC void\* MC\_GetAgentExecEngine (MCAgent\_t *agent*)**

Retrieve an agent's Ch interpreter.

**Parameters:**

*agent* a valid agent

**Returns:**

a Ch interpreter of type 'ChInterp\_t' on success, or NULL on failure.

Definition at line 1066 of file libmc.c.

References agent\_s::agent\_interp.

**13.49.12.5 EXPORTMC int MC\_GetAgentID (MCAgent\_t *agent*)**

Retrieve an agent's id.

Definition at line 1072 of file libmc.c.

References agent\_s::id.

Referenced by MC\_GetAgentID\_chdl().

**13.49.12.6 EXPORTCH int MC\_GetAgentID\_chdl (void \* *varg*)**

Definition at line 2858 of file libmc.c.

References MC\_GetAgentID().

Referenced by agent\_ChScriptInitVar().

**13.49.12.7 EXPORTMC char\* MC\_GetAgentName (MCAgent\_t *agent*)**

Definition at line 1080 of file libmc.c.

References agent\_s::lock, MUTEX\_LOCK, MUTEX\_UNLOCK, and agent\_s::name.

Referenced by MC\_GetAgentName\_chdl().

**13.49.12.8 EXPORTCH char\* MC\_GetAgentName\_chdl (void \* *varg*)**

Definition at line 2874 of file libmc.c.

References MC\_GetAgentName().

Referenced by agent\_ChScriptInitVar().

**13.49.12.9 EXPORTMC int MC\_GetAgentNumTasks (MCAgent\_t *agent*)**

Retrive the number of tasks an agent has.

**13.49.13 Example**

Definition at line 1161 of file libmc.c.

References agent\_s::datastate, and agent\_datastate\_s::number\_of\_tasks.

Referenced by MC\_GetAgentNumTasks\_chdl().

**13.49.13.1 EXPORTCH int MC\_GetAgentNumTasks\_chdl (void \* *varg*)**

Definition at line 2890 of file libmc.c.

References MC\_GetAgentNumTasks().

### 13.49.13.2 EXPORTMC int MC\_GetAgentReturnData (MCAgent\_t *agent*, int *task\_num*, void \*\**data*, int \**dim*, int \*\**extent*)

Get an agent's return data.

#### Parameters:

*agent* a valid agent

*task\_num* the task for which to retrieve the return data. The task must already be completed.

*data* the return data. May be multi dimensional array.

*dim* the number of dimensions of the return array.

*extent* the extent of each one of the array dimensions.

### 13.49.14 Example

This file demonstrates the retrieval of agent return data from an agent

This is the agent which gets the data

Definition at line 1098 of file libmc.c.

References agent\_task\_s::agent\_return\_data, interpreter\_variable\_data\_s::array\_dim, interpreter\_variable\_data\_s::array\_extent, CH\_DATATYPE\_SIZE, interpreter\_variable\_data\_s::data\_type, agent\_s::datastate, agent\_datastate\_s::number\_of\_tasks, size, and agent\_datastate\_s::tasks.

#### 13.49.14.1 EXPORTMC int MC\_GetAgentStatus (MCAgent\_t *agent*)

Get an agent's current status.

#### Returns:

returns type 'enum MC\_AgentStatus\_e'

Definition at line 1044 of file libmc.c.

References agent\_s::agent\_status, agent\_s::lock, MUTEX\_LOCK, and MUTEX\_UNLOCK.

Referenced by MC\_GetAgentStatus\_chdl().

#### 13.49.14.2 EXPORTCH int MC\_GetAgentStatus\_chdl (void \* *varg*)

Definition at line 2906 of file libmc.c.

References MC\_GetAgentStatus().

Referenced by agent\_ChScriptInitVar().

**13.49.14.3 EXPORTMC enum MC\_AgentType\_e MC\_GetAgentType (MCAgent\_t *agent*)**

Get an agent's type.

**Returns:**

returns type 'enum MC\_AgentType\_e'

Definition at line 1167 of file libmc.c.

References agent\_s::agent\_type.

**13.49.14.4 EXPORTMC char\* MC\_GetAgentXMLString (MCAgent\_t *agent*)**

Get an agent's xml string.

**Returns:**

a malloc'd character string containing the agent's xml code

Definition at line 1054 of file libmc.c.

References agent\_s::datastate, mxmlSaveAllocString(), and agent\_datastate\_s::xml\_agent\_root.

Referenced by MC\_GetAgentXMLString\_chdl().

**13.49.14.5 EXPORTCH char\* MC\_GetAgentXMLString\_chdl (void \* *varg*)**

Definition at line 2922 of file libmc.c.

References MC\_GetAgentXMLString().

Referenced by agent\_ChScriptInitVar().

**13.49.14.6 int MC\_GetAllAgents (MCAgency\_t *attr*, MCAgent\_t \*\* *agents*, int \* *num\_agents*)**

Definition at line 1177 of file libmc.c.

References mc\_platform\_s::agent\_queue, mc\_platform\_s::giant, mc\_platform\_s::giant\_lock, MC\_HaltAgency(), agency\_s::mc\_platform, MC\_ResumeAgency(), MUTEX\_LOCK, and MUTEX\_UNLOCK.

**13.49.14.7 EXPORTCH int MC\_GetTimeOfDay\_chdl (void \* *varg*)**

Definition at line 2938 of file libmc.c.

Referenced by agent\_ChScriptInitVar().

**13.49.14.8 EXPORTMC int MC\_HaltAgency (MCAgency\_t *agency*)**

Halt an agency: Do not process new entries in queues.

**Parameters:**

*agency* A handle to a running MobileC agency.

**Returns:**

0 on success, non-zero on failure.

Definition at line 1213 of file libmc.c.

References `mc_platform_s::giant`, `mc_platform_s::giant_lock`, `agency_s::mc_platform`, `MUTEX_LOCK`, and `MUTEX_UNLOCK`.

Referenced by `MC_GetAllAgents()`, and `MC_HaltAgency_chdl()`.

**13.49.14.9 EXPORTCH int MC\_HaltAgency\_chdl (void \* *varg*)**

Definition at line 2953 of file libmc.c.

References `CHECK_NULL`, `MC_HaltAgency()`, and `agency_s::mc_platform`.

Referenced by `agent_ChScriptInitVar()`.

**13.49.14.10 EXPORTMC MCAgency\_t MC\_Initialize (int *port*, MCAgencyOptions\_t \* *options*)**

Initialize and start a MobileC agency.

**Parameters:**

*port* the TCP port the agency should bind to

*options* initialized MobileC [options](#) or NULL for default [options](#)

**Returns:**

a handle to a running MobileC agency or NULL on failure

**13.49.15 Example**

Definition at line 1222 of file libmc.c.

References `mc_platform_s::agency`, `buf`, `MCAgencyOptions_s::ch_options`, `CHECK_NULL`, `agency_s::client`, `MCAgencyOptions_s::default_agent_status`, `agency_s::default_agentstatus`, `f`, `HOST_NAME_MAX`, `agency_s::hostname`, `MCAgencyOptions_s::initInterps`, `agency_s::initInterps`, `agency_s::known_host_filename`, `MCAgencyOptions_s::known_host_filename`, `MC_InitializeAgencyOptions()`, `mc_platform`, `agency_s::mc_platform`, `mc_platform_Initialize()`, `MC_THREAD_ALL`, `MCAgencyOptions_s::passphrase`, `agency_s::portno`, `agency_s::priv_key_filename`, `MCAgencyOptions_s::priv_key_filename`, `read_encrypted_file()`, `agency_s::server`, `MCAgencyOptions_s::stack_size`, `agency_s::stack_size`, `MCAgencyOptions_s::threads`, and `agency_s::threads`.

**13.49.15.1 EXPORTMC int MC\_InitializeAgencyOptions (struct MCAgencyOptions\_s \* *options*)**

Initialize MobileC [options](#).

**Parameters:**

*options* [options](#) to initialize.

**Returns:**

0 on success, `error_code_t` on failure

**Note:**

MobileC [options](#) should be initialized with this function before any of its members are modified.

**13.49.16 Example**

Definition at line 1333 of file `libmc.c`.

References `MCAgencyOptions_s::default_agent_status`, `MCAgencyOptions_s::initInterps`, `MCAgencyOptions_s::known_host_filename`, `MC_THREAD_ALL`, `MC_WAIT_CH`, `MCAgencyOptions_s::modified`, `MCAgencyOptions_s::passphrase`, `MCAgencyOptions_s::priv_key_filename`, `MCAgencyOptions_s::stack_size`, and `MCAgencyOptions_s::threads`.

Referenced by `MC_Initialize()`.

**13.49.16.1 EXPORTMC int MC\_LoadAgentFromFile (MCAgency\_t attr, const char \* filename)**

Load an agent from a file into an agency.

**Parameters:**

*agency* A valid and running Mobile-C agency

*filename* Filename containing the agent to load

**Returns:**

0 on success, non-zero on failure.

Definition at line 1354 of file `libmc.c`.

References `buf`, `agency_s::mc_platform`, `message_Destroy()`, `message_InitializeFromString()`, `message_New()`, `mc_platform_s::message_queue`, `MXML_DESCEND`, `mxmlFindElement()`, `mxmlLoadString()`, `message_s::to_address`, `message_s::xml_payload`, and `message_s::xml_root`.

**13.49.16.2 int MC\_MainLoop (MCAgency\_t attr)**

Wait indefinitely.

**Note:**

This function is intended to block the calling thread forever.

Definition at line 2008 of file `libmc.c`.

References `COND_WAIT`, `agency_s::mc_platform`, `MUTEX_LOCK`, `MUTEX_UNLOCK`, `mc_platform_s::quit`, `mc_platform_s::quit_cond`, and `mc_platform_s::quit_lock`.



### 13.49.16.3 EXPORTMC int MC\_MigrateAgent (MCAgent\_t *agent*, const char \* *hostname*, int *port*)

Migrates a running agent to another host.

#### Parameters:

*agent* The agent to migrate

*hostname* The new host to migrate the agent to

*port* The new port to migrate the agent to

#### Returns:

0 on success, error\_code\_t type on failure.

Definition at line 1424 of file libmc.c.

References agent\_s::datastate, agent\_datastate\_s::progress\_modifier, agent\_task\_s::server\_name, agent\_datastate\_s::task\_progress, and agent\_datastate\_s::tasks.

Referenced by MC\_MigrateAgent\_chdl().

### 13.49.16.4 EXPORTCH int MC\_MigrateAgent\_chdl (void \* *varg*)

Definition at line 2970 of file libmc.c.

References MC\_MigrateAgent(), and port.

Referenced by agent\_ChScriptInitVar().

### 13.49.16.5 EXPORTMC int MC\_MutexLock (MCAgency\_t *attr*, int *id*)

Locks a MobileC synchronization variable as a mutex.

#### Parameters:

*attr* a MobileC agency handle

*id* the synchronization variable id to lock

#### Returns:

0 on success, error\_code\_t type on failure

### 13.49.17 Example

Consider the following agents, which use the agent-space version of this api function. Note that the 'sleep' agent is sent first, followed by the 'wake' agent.

Definition at line 1446 of file libmc.c.

References syncListNode\_s::lock, agency\_s::mc\_platform, MUTEX\_LOCK, mc\_platform\_s::syncList, and syncListFind().

Referenced by MC\_MutexLock\_chdl().

**13.49.17.1 EXPORTCH int MC\_MutexLock\_chdl (void \* *varg*)**

Definition at line 2989 of file libmc.c.

References CHECK\_NULL, MC\_MutexLock(), and agency\_s::mc\_platform.

Referenced by agent\_ChScriptInitVar().

**13.49.17.2 EXPORTMC int MC\_MutexUnlock (MCAgency\_t *attr*, int *id*)**

Definition at line 1458 of file libmc.c.

References syncListNode\_s::lock, agency\_s::mc\_platform, MUTEX\_UNLOCK, mc\_platform\_s::syncList, and syncListFind().

Referenced by MC\_MutexUnlock\_chdl().

**13.49.17.3 EXPORTCH int MC\_MutexUnlock\_chdl (void \* *varg*)**

Definition at line 3013 of file libmc.c.

References CHECK\_NULL, MC\_MutexUnlock(), and agency\_s::mc\_platform.

Referenced by agent\_ChScriptInitVar().

**13.49.17.4 EXPORTMC int MC\_PrintAgentCode (MCAgent\_t *agent*)**

Prints an agents code to stdout.

**Returns:**

0 on success, error\_code\_t on failure

Definition at line 1470 of file libmc.c.

References agent\_datastate\_s::agent\_code, agent\_s::datastate, agent\_s::lock, MUTEX\_LOCK, MUTEX\_UNLOCK, agent\_datastate\_s::number\_of\_tasks, and agent\_datastate\_s::task\_progress.

Referenced by MC\_PrintAgentCode\_chdl().

**13.49.17.5 EXPORTCH int MC\_PrintAgentCode\_chdl (void \* *varg*)**

Definition at line 3037 of file libmc.c.

References MC\_PrintAgentCode().

Referenced by agent\_ChScriptInitVar().

**13.49.17.6 EXPORTMC int MC\_RegisterService (MCAgency\_t *agency*, MCAgent\_t *agent*, int *agentID*, const char \* *agentName*, char \*\* *serviceNames*, int *numServices*)**

Register a new service with the Directory Facilitator.

**Parameters:**

*agency* a MobileC agency handle

*agent* (OPTIONAL: See note) a MobileC agent  
*agentID* (OPTIONAL: See note) a MobileC agent id  
*agentName* (OPTIONAL: See note) a MobileC agent name  
*serviceNames* an array of character strings of service names  
*numServices* the number of services described in 'serviceNames'

**Returns:**

0 on success, error\_code\_t type on failure

**Note:**

Three of the input arguments are optional. The function expects as input the arguments 'agent XOR (agentID AND agentName)'.

### 13.49.18 Example

Definition at line 1486 of file libmc.c.

References CHECK\_NULL, mc\_platform\_s::df, df\_AddRequest(), df\_request\_list\_node\_New(), agent\_s::id, MC\_ERR\_INVALID\_ARGS, MC\_ERR\_MEMORY, agency\_s::mc\_platform, MUTEX\_INIT, MUTEX\_T, and agent\_s::name.

Referenced by MC\_RegisterService\_chdl().

#### 13.49.18.1 EXPORTCH int MC\_RegisterService\_chdl (void \* varg)

Definition at line 3053 of file libmc.c.

References CHECK\_NULL, agency\_s::mc\_platform, and MC\_RegisterService().

Referenced by agent\_ChScriptInitVar().

#### 13.49.18.2 EXPORTMC int MC\_ResetSignal (MCAgency\_t attr)

Reset a MobileC signal.

**Returns:**

0 on success, error\_code\_t on failure

**See also:**

[MC\\_WaitSignal\(\)](#)

Definition at line 1618 of file libmc.c.

References COND\_SIGNAL, mc\_platform\_s::giant, mc\_platform\_s::giant\_cond, mc\_platform\_s::giant\_lock, MC\_NO\_SIGNAL, agency\_s::mc\_platform, mc\_platform\_s::MC\_signal, MUTEX\_LOCK, and MUTEX\_UNLOCK.

**13.49.18.3 EXPORTMC int MC\_ResumeAgency (MCAgency\_t *agency*)**

Resumes a halted agency.

**Parameters:**

*agency* An agency previously halted with the [MC\\_HaltAgency\(\)](#) function.

**Returns:**

0 on success, non-zero on failure.

Definition at line 1565 of file libmc.c.

References mc\_platform\_s::giant, mc\_platform\_s::giant\_lock, agency\_s::mc\_platform, MUTEX\_LOCK, and MUTEX\_UNLOCK.

Referenced by MC\_GetAllAgents(), and MC\_ResumeAgency\_chdl().

**13.49.18.4 EXPORTCH int MC\_ResumeAgency\_chdl (void \* *varg*)**

Definition at line 3090 of file libmc.c.

References CHECK\_NULL, agency\_s::mc\_platform, and MC\_ResumeAgency().

Referenced by agent\_ChScriptInitVar().

**13.49.18.5 EXPORTMC MCAgent\_t MC\_RetrieveAgent (MCAgency\_t *attr*)**

Retrieves the oldest agent from an agency.

**Returns:**

a valid agent or NULL on failure

Definition at line 1574 of file libmc.c.

References mc\_platform\_s::agent\_queue, agent\_s::agent\_status, ListSearch(), MC\_AGENT\_NEUTRAL, agency\_s::mc\_platform, MUTEX\_LOCK, and MUTEX\_UNLOCK.

Referenced by MC\_RetrieveAgent\_chdl().

**13.49.18.6 EXPORTCH MCAgent\_t MC\_RetrieveAgent\_chdl (void \* *varg*)**

Definition at line 3108 of file libmc.c.

References CHECK\_NULL, agency\_s::mc\_platform, and MC\_RetrieveAgent().

Referenced by agent\_ChScriptInitVar().

**13.49.18.7 EXPORTMC char\* MC\_RetrieveAgentCode (MCAgent\_t *agent*)**

Retrieves an agent's Ch code.

**Returns:**

a malloc'd character string on success, NULL on failure

Definition at line 1602 of file libmc.c.

References `agent_datastate_s::agent_code`, `buf`, `agent_s::datastate`, `agent_s::lock`, `MUTEX_LOCK`, `MUTEX_UNLOCK`, and `agent_datastate_s::task_progress`.

Referenced by `MC_RetrieveAgentCode_chdl()`.

#### 13.49.18.8 EXPORTCH char\* MC\_RetrieveAgentCode\_chdl (void \* *varg*)

Definition at line 3125 of file libmc.c.

References `MC_RetrieveAgentCode()`.

Referenced by `agent_ChScriptInitVar()`.

#### 13.49.18.9 EXPORTCH int MC\_SaveData\_chdl (void \* *varg*)

Definition at line 3141 of file libmc.c.

References `agent_task_s::agent_variable_list`, `interpreter_variable_data_s::data`, `interpreter_variable_data_s::data_type`, `agent_s::datastate`, `interpreter_variable_data_New()`, `interpreter_variable_data_s::name`, `interpreter_variable_data_s::size`, `size`, `agent_datastate_s::task_progress`, and `agent_datastate_s::tasks`.

Referenced by `agent_ChScriptInitVar()`.

#### 13.49.18.10 EXPORTMC int MC\_SearchForService (MC\_Agency\_t *attr*, const char \* *searchString*, char \*\*\* *agentNames*, char \*\*\* *serviceNames*, int \*\* *agentIDs*, int \* *numResults*)

Search the directory facilitator for a service.

##### Returns:

0 on success, `error_code_t` on failure

##### Parameters:

*attr* (input) a MobileC agency handle

*searchString* (input) substring to search services for

*agentNames* (return) array of agent names with matching services

*serviceNames* (return) array of matching service names

*agentIDs* (return) array of matching agent IDs

*numResults* (return) number of matching results

#### 13.49.19 Example

Definition at line 1629 of file libmc.c.

References `CHECK_NULL`, `COND_SLEEP_ACTION`, `mc_platform_s::df`, `df_AddRequest()`, `df_request_list_node_Destroy()`, `df_request_list_node_New()`, `df_request_search_Destroy()`, `df_request_search_New()`, `MC_ERR_MEMORY`, `agency_s::mc_platform`, `MC_SUCCESS`, and `search`.

Referenced by `MC_SearchForService_chdl()`.

**13.49.19.1 EXPORTCH int MC\_SearchForService\_chdl (void \* *varg*)**

Definition at line 3176 of file libmc.c.

References CHECK\_NULL, agency\_s::mc\_platform, and MC\_SearchForService().

Referenced by agent\_ChScriptInitVar().

**13.49.19.2 EXPORTMC int MC\_SemaphorePost (MCAgency\_t *attr*, int *id*)**

Post to a MobileC synchronization variable semaphore.

**Parameters:**

*attr* a MobileC agency handle

*id* the synchronization variable id to post to

**Returns:**

0 on success, error\_code\_t type on failure

**13.49.20 Example**

Definition at line 1679 of file libmc.c.

References agency\_s::mc\_platform, syncListNode\_s::sem, SEMAPHORE\_POST, mc\_platform\_s::syncList, and syncListFind().

Referenced by MC\_SemaphorePost\_chdl().

**13.49.20.1 EXPORTCH int MC\_SemaphorePost\_chdl (void \* *varg*)**

Definition at line 3217 of file libmc.c.

References CHECK\_NULL, agency\_s::mc\_platform, and MC\_SemaphorePost().

Referenced by agent\_ChScriptInitVar().

**13.49.20.2 EXPORTMC int MC\_SemaphoreWait (MCAgency\_t *attr*, int *id*)**

Decreases a MobileC synchronization variable semaphore count by one.

**Parameters:**

*attr* a MobileC agency handle

*id* synchronization variable id to wait on

**Returns:**

0 on MC\_SUCCESS, error\_code\_t type of failure

**Note:**

If the semaphore count is already zero, this function will block until another thread posts to the semaphore.

### 13.49.21 Example

Definition at line 1691 of file libmc.c.

References `agency_s::mc_platform`, `syncListNode_s::sem`, `SEMAPHORE_WAIT`, `mc_platform_s::syncList`, and `syncListFind()`.

Referenced by `MC_SemaphoreWait_chdl()`.

#### 13.49.21.1 EXPORTCH int MC\_SemaphoreWait\_chdl (void \* *varg*)

Definition at line 3241 of file libmc.c.

References `CHECK_NULL`, `agency_s::mc_platform`, and `MC_SemaphoreWait()`.

Referenced by `agent_ChScriptInitVar()`.

#### 13.49.21.2 EXPORTMC int MC\_SendAgentMigrationMessage (MCAgency\_t *attr*, const char \* *message*, const char \* *hostname*, int *port*)

Sends an agent migration message.

##### Parameters:

- attr* a MobileC agency handle
- message* a valid MobileC xml agent migration message
- hostname* host to send the message to
- port* port to send the message to

Definition at line 1713 of file libmc.c.

References `MC_ERR`, `agency_s::mc_platform`, `message_Destroy()`, `message_InitializeFromString()`, `message_New()`, and `mc_platform_s::message_queue`.

Referenced by `MC_SendAgentMigrationMessage_chdl()`.

#### 13.49.21.3 EXPORTCH int MC\_SendAgentMigrationMessage\_chdl (void \* *varg*)

Definition at line 3265 of file libmc.c.

References `CHECK_NULL`, `agency_s::mc_platform`, `MC_SendAgentMigrationMessage()`, and `port`.

Referenced by `agent_ChScriptInitVar()`.

#### 13.49.21.4 EXPORTMC int MC\_SendAgentMigrationMessageFile (MCAgency\_t *attr*, const char \* *filename*, const char \* *hostname*, int *port*)

Sends an agent migration message.

##### Parameters:

- attr* a MobileC agency handle
- filename* file containing a valid MobileC xml agent migration message

**hostname** hostname to send the agent to

**port** port to send the agent to

Definition at line 1744 of file libmc.c.

References agent\_Initialize(), mc\_platform\_s::agent\_queue, agent\_s::agent\_status, mc\_platform\_s::ams, buf, COND\_BROADCAST, agency\_s::mc\_platform, MC\_WAIT\_MESSGSEND, message\_Destroy(), message\_InitializeFromString(), message\_New(), MUTEX\_LOCK, and MUTEX\_UNLOCK.

Referenced by handler\_SEND(), and MC\_SendAgentMigrationMessageFile\_chdl().

### 13.49.21.5 EXPORTCH int MC\_SendAgentMigrationMessageFile\_chdl (void \* *varg*)

Definition at line 3290 of file libmc.c.

References MC\_SendAgentMigrationMessageFile(), and port.

Referenced by agent\_ChScriptInitVar().

### 13.49.21.6 int MC\_SendCh (MCAgency\_t *attr*, const char \* *filename*, const char \* *remotehost*, int *port*)

Definition at line 1703 of file libmc.c.

### 13.49.21.7 EXPORTMC int MC\_SendSteerCommand (MCAgency\_t *attr*, enum MC\_SteerCommand\_e *cmd*)

Definition at line 1812 of file libmc.c.

References COND\_BROADCAST, agency\_s::mc\_platform, mc\_platform\_s::MC\_steer\_command, mc\_platform\_s::MC\_steer\_cond, mc\_platform\_s::MC\_steer\_lock, MUTEX\_LOCK, and MUTEX\_UNLOCK.

Referenced by MC\_SendSteerCommand\_chdl().

### 13.49.21.8 EXPORTCH int MC\_SendSteerCommand\_chdl (void \* *varg*)

Definition at line 3310 of file libmc.c.

References CHECK\_NULL, agency\_s::mc\_platform, and MC\_SendSteerCommand().

Referenced by agent\_ChScriptInitVar().

### 13.49.21.9 int MC\_SetAgentStatus (MCAgent\_t *agent*, int *status*)

Set an agent's status.

#### Parameters:

**agent** a MobileC agent

**status** agent status of type 'enum MC\_AgentStatus\_e'

#### Returns:

0 on success, or error\_code\_t on failure



Definition at line 1822 of file libmc.c.

References `agent_s::agent_status`, `mc_platform_s::ams`, `COND_SIGNAL`, `agent_s::lock`, `agent_s::mc_platform`, `MUTEX_LOCK`, `MUTEX_UNLOCK`, and `agent_s::orphan`.

Referenced by `MC_DeleteAgent()`, `MC_DeleteAgentWG()`, and `MC_SetAgentStatus_chdl()`.

#### 13.49.21.10 EXPORTCH int MC\_SetAgentStatus\_chdl (void \* *varg*)

Definition at line 3333 of file libmc.c.

References `MC_SetAgentStatus()`.

Referenced by `agent_ChScriptInitVar()`.

#### 13.49.21.11 int MC\_SetDefaultAgentStatus (MCAgency\_t *agency*, enum MC\_AgentStatus\_e *status*)

Sets default incoming agent status.

##### Parameters:

*agency* a MobileC agency handle

*status* the status to set all incoming agents

##### Returns:

0 on success, `error_type_t` on failure

##### Note:

using this function will override any status the incoming agent attempts to set for itself.

Definition at line 1837 of file libmc.c.

References `mc_platform_s::default_agentstatus`, and `agency_s::mc_platform`.

Referenced by `MC_SetDefaultAgentStatus_chdl()`.

#### 13.49.21.12 EXPORTCH int MC\_SetDefaultAgentStatus\_chdl (void \* *varg*)

Definition at line 3351 of file libmc.c.

References `CHECK_NULL`, `agency_s::mc_platform`, and `MC_SetDefaultAgentStatus()`.

Referenced by `agent_ChScriptInitVar()`.

#### 13.49.21.13 EXPORTMC int MC\_SetThreadOff (MCAgencyOptions\_t \* *options*, enum MC\_ThreadIndex\_e *index*)

Sets a MobileC thread to "off" status.

##### Parameters:

*options* MobileC [options](#) previously initialized with [MC\\_InitializeAgencyOptions\(\)](#)

*index* the thread to set

**Returns:**

0 on success, `error_code_t` on failure

**Note:**

This function must be called before `MC_Initialize()`. Once an agency is started with `MC_Initialize`, the `MC_SetThread` functions will have no effect.

Definition at line 1864 of file `libmc.c`.

References `SET_THREAD_OFF`, and `MCAgencyOptions_s::threads`.

**13.49.21.14 EXPORTMC int MC\_SetThreadOn (MCAgencyOptions\_t \* *options*, enum MC\_ThreadIndex\_e *index*)**

Sets a MobileC thread to "on" status.

**Parameters:**

*options* MobileC *options* previously initialized with `MC_InitializeAgencyOptions()`

*index* the thread to set

**Returns:**

0 on success, `error_code_t` on failure

**Note:**

This function must be called before `MC_Initialize()`. Once an agency is started with `MC_Initialize`, the `MC_SetThread` functions will have no effect.

Definition at line 1847 of file `libmc.c`.

References `SET_THREAD_ON`, and `MCAgencyOptions_s::threads`.

**13.49.21.15 EXPORTMC int MC\_SetThreadsAllOff (MCAgencyOptions\_t \* *options*)**

Set all MobileC threads to 'off' status.

**Parameters:**

*options* a MobileC *options* structure initialized with with the `MC_InitializeAgencyOptions()` function.

**Returns:**

0 on success, error code on failure.

Definition at line 1871 of file `libmc.c`.

References `MC_THREAD_ALL`, `SET_THREAD_OFF`, and `MCAgencyOptions_s::threads`.

**13.49.21.16 EXPORTMC int MC\_SetThreadsAllOn (MCAgencyOptions\_t \* *options*)**

Set all Mobile-C threads on.

**Parameters:**

*options* MobileC [options](#) structure, initialized with [MC\\_InitializeAgencyOptions\(\)](#)

**Returns:**

0 on success, error code on failure.

Definition at line 1854 of file libmc.c.

References [MC\\_THREAD\\_ALL](#), [SET\\_THREAD\\_ON](#), and [MCAgencyOptions\\_s::threads](#).

**13.49.21.17 EXPORTMC int MC\_Steer (MCAgency\_t *attr*, int(\*) (void \**data*) *funcptr*, void \**arg*)**

Set up a steerable algorithm.

**Parameters:**

*attr* a MobileC agency handle

*funcptr* a function pointer to the algorithm

*arg* an argument for the algorithm function

**Returns:**

0 on success, [error\\_code\\_t](#) on failure

**Note:**

The algorithm function must contain a call to [MC\\_SteerControl](#) in order for the algorithm to be steerable.

**13.49.22 Example**

Definition at line 1881 of file libmc.c.

References [agency\\_s::mc\\_platform](#), [MC\\_RESTART](#), [MC\\_RUN](#), [mc\\_platform\\_s::MC\\_steer\\_command](#), [mc\\_platform\\_s::MC\\_steer\\_lock](#), [MUTEX\\_LOCK](#), and [MUTEX\\_UNLOCK](#).

**13.49.22.1 EXPORTMC enum MC\_SteerCommand\_e MC\_SteerControl (void)**

The MobileC user-algorithm steering function.

**Returns:**

The current steering command

**Note:**

This function belongs inside a user's steerable algorithm.

**See also:**

[MC\\_Steer\(\)](#)

Definition at line 1900 of file libmc.c.

References COND\_WAIT, mc\_platform\_s::MC\_steer\_command, mc\_platform\_s::MC\_steer\_cond, mc\_platform\_s::MC\_steer\_lock, MC\_SUSPEND, MUTEX\_LOCK, and MUTEX\_UNLOCK.

### 13.49.22.2 EXPORTMC int MC\_SyncDelete (MCAgency\_t *attr*, int *id*)

Deletes a previously initialized synchronization variable.

#### Parameters:

*attr* a MobileC agency handle

*id* the sync variable id to delete

#### Returns:

0 on success, or error\_code\_t on failure

Definition at line 1916 of file libmc.c.

References syncList\_s::giant\_lock, syncListNode\_s::lock, MC\_ERR\_NOT\_FOUND, agency\_s::mc\_platform, MUTEX\_LOCK, MUTEX\_UNLOCK, mc\_platform\_s::syncList, syncListFind(), syncListNodeDestroy(), and syncListRemove().

Referenced by MC\_SyncDelete\_chdl().

### 13.49.22.3 EXPORTCH int MC\_SyncDelete\_chdl (void \* *varg*)

Definition at line 3374 of file libmc.c.

References CHECK\_NULL, agency\_s::mc\_platform, and MC\_SyncDelete().

Referenced by agent\_ChScriptInitVar().

### 13.49.22.4 EXPORTMC int MC\_SyncInit (MCAgency\_t *attr*, int *id*)

Initializes a new MobileC synchronization variable.

#### Parameters:

*attr* a MobileC agency handle

*id* the requested sync variable id

#### Returns:

new sync variable's id. May be different than the requested id.

#### Note:

Each synchronization variable may be used as a mutex, condition variable, or semaphore. However, it should only be used as one type of synchronization variable per instance, or undefined behaviour may result.

#### See also:

[MC\\_MutexLock\(\)](#), [MC\\_MutexUnlock\(\)](#), [MC\\_CondWait\(\)](#), [MC\\_CondSignal\(\)](#), [MC\\_CondBroadcast](#), [MC\\_SemaphorePost\(\)](#), [MC\\_SemaphoreWait\(\)](#)

Definition at line 1946 of file libmc.c.

References syncList\_s::giant\_lock, syncListNode\_s::id, agency\_s::mc\_platform, MUTEX\_LOCK, MUTEX\_UNLOCK, node, mc\_platform\_s::syncList, syncListAddNode(), syncListFind(), and syncListNodeNew().

Referenced by MC\_SyncInit\_chdl().

#### 13.49.22.5 EXPORTCH int MC\_SyncInit\_chdl (void \* *varg*)

Definition at line 3397 of file libmc.c.

References CHECK\_NULL, agency\_s::mc\_platform, and MC\_SyncInit().

Referenced by agent\_ChScriptInitVar().

#### 13.49.22.6 EXPORTMC int MC\_TerminateAgent (MCAgent\_t *agent*)

Halt a running agent.

##### Returns:

0 on success, error\_code\_t on failure

Definition at line 1971 of file libmc.c.

References agent\_s::agent\_interp.

Referenced by ams\_ManageAgentList(), MC\_DeleteAgent(), and MC\_TerminateAgent\_chdl().

#### 13.49.22.7 EXPORTCH int MC\_TerminateAgent\_chdl (void \* *varg*)

Definition at line 3421 of file libmc.c.

References MC\_ERR\_NOT\_FOUND, MC\_FindAgentByName(), and MC\_TerminateAgent().

Referenced by agent\_ChScriptInitVar().

#### 13.49.22.8 EXPORTMC int MC\_TerminateAgentWG (MCAgent\_t *calling\_agent*, MCAgent\_t *agent*)

Definition at line 1981 of file libmc.c.

References agent\_s::agent\_interp, MC\_ERR\_INVALID\_ARGS, and agent\_s::wg\_code.

Referenced by MC\_DeleteAgentWG(), and MC\_TerminateAgentWG\_chdl().

#### 13.49.22.9 EXPORTCH int MC\_TerminateAgentWG\_chdl (void \* *varg*)

Definition at line 3442 of file libmc.c.

References MC\_ERR\_NOT\_FOUND, MC\_FindAgentByName(), and MC\_TerminateAgentWG().

Referenced by agent\_ChScriptInitVar().

**13.49.22.10 EXPORTMC int MC\_WaitAgent (MCAgency\_t *attr*)**

Wait indefinitely.

**Note:**

This function is intended to block the calling thread forever. Wait for an agent arrival event  
This function blocks until an agent arrival signal is triggered, at which point it unblocks.

Definition at line 2020 of file libmc.c.

References `mc_platform_s::agent_queue`, `COND_WAIT`, `mc_platform`, `agency_s::mc_platform`, `MUTEX_LOCK`, `MUTEX_UNLOCK`, and `size`.

**13.49.22.11 EXPORTMC MCAgent\_t MC\_WaitRetrieveAgent (MCAgency\_t *attr*)**

Wait and retrieve an agent.

**Returns:**

a valid MobileC agent on success, or NULL on failure

**Note:**

This function blocks until the arrival of an agent. The agent is retrieved after it is initialized, but before it is executed.

Definition at line 2040 of file libmc.c.

References `mc_platform_s::agent_queue`, `ListSearch()`, `agency_s::mc_platform`, `MC_RECV_AGENT`, `MC_WaitSignal()`, `MUTEX_LOCK`, and `MUTEX_UNLOCK`.

**13.49.22.12 EXPORTMC int MC\_WaitSignal (MCAgency\_t *attr*, int *signals*)**

Wait for a MobileC signal.

**Parameters:**

*attr* a MobileC agency handle

*signals* a flag of signals to wait for, of type 'enum MC\_Signal\_e'

**Returns:**

0 on success, `error_code_t` on failure

**Note:**

the parameter 'signals' may be something like 'MC\_RECV\_MESSAGE | MC\_RECV\_AGENT', etc.

**13.49.23 Example**

Definition at line 2058 of file libmc.c.

References COND\_WAIT, mc\_platform\_s::giant, mc\_platform\_s::giant\_lock, agency\_s::mc\_platform, mc\_platform\_s::MC\_signal, mc\_platform\_s::MC\_signal\_cond, mc\_platform\_s::MC\_signal\_lock, MUTEX\_LOCK, and MUTEX\_UNLOCK.

Referenced by MC\_WaitRetrieveAgent().

## 13.49.24 Variable Documentation

### 13.49.24.1 mc\_platform\_p g\_mc\_platform

Definition at line 71 of file libmc.c.

Referenced by fipa\_envelope\_Compose\_\_from().

## 13.50 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mc\_list/list.c File Reference

```
#include "list.h"
#include <stdio.h>
#include <stdlib.h>
```

### Functions

- [list\\_p ListInitialize](#) (void)
- void [ListTerminate](#) ([list\\_p](#) list)
- int [list\\_pGetSize](#) ([list\\_p](#) list)
- DATA [ListGetHead](#) ([list\\_p](#) list)
- DATA [ListPop](#) ([list\\_p](#) list)
- DATA [ListSearch](#) ([list\\_p](#) list, const int index)
- int [ListAdd](#) ([list\\_p](#) list, DATA data)
- int [ListInsert](#) ([list\\_p](#) list, DATA data, const int index)
- DATA [ListDelete](#) ([list\\_p](#) list, const int index)

### 13.50.1 Function Documentation

#### 13.50.1.1 int list\_pGetSize ([list\\_p](#) list)

Definition at line 73 of file list.c.

References [list\\_s::size](#).

#### 13.50.1.2 int ListAdd ([list\\_p](#) list, DATA data)

Definition at line 161 of file list.c.

References [list\\_s::listhead](#), [listNode\\_s::next](#), [listNode\\_s::node\\_data](#), and [list\\_s::size](#).

Referenced by [agent\\_task\\_Copy\(\)](#), [barrier\\_queue\\_Add\(\)](#), [df\\_Add\(\)](#), [df\\_AddRequest\(\)](#), and [syncListAddNode\(\)](#).

#### 13.50.1.3 DATA ListDelete ([list\\_p](#) list, const int index)

Definition at line 216 of file list.c.

References [DATA](#), [list\\_s::listhead](#), [listNode\\_s::next](#), [listNode\\_s::node\\_data](#), and [list\\_s::size](#).

Referenced by [barrier\\_queue\\_Delete\(\)](#), [syncListDelete\(\)](#), and [syncListRemove\(\)](#).

#### 13.50.1.4 DATA ListGetHead ([list\\_p](#) list)

Definition at line 79 of file list.c.

References [list\\_s::listhead](#), and [listNode\\_s::node\\_data](#).



#### 13.50.1.5 list\_p ListInitialize (void)

Definition at line 33 of file list.c.

References list\_s::listhead, and list\_s::size.

Referenced by barrier\_queue\_New(), df\_Initialize(), df\_request\_list\_New(), and syncListInit().

#### 13.50.1.6 int ListInsert (list\_p list, DATA data, const int index)

Definition at line 203 of file list.c.

#### 13.50.1.7 DATA ListPop (list\_p list)

Definition at line 89 of file list.c.

References DATA, list\_s::listhead, listNode\_s::next, listNode\_s::node\_data, and list\_s::size.

Referenced by AP\_QUEUE\_STD\_DEFN\_TEMPLATE(), barrier\_queue\_Pop(), df\_Destroy(), df\_request\_list\_Destroy(), and df\_request\_list\_Pop().

#### 13.50.1.8 DATA ListSearch (list\_p list, const int index)

Definition at line 124 of file list.c.

References list\_s::listhead, listNode\_s::next, listNode\_s::node\_data, and list\_s::size.

Referenced by agent\_queue\_Flush(), agent\_task\_Copy(), ams\_ManageAgentList(), ams\_Print(), AP\_QUEUE\_SEARCH\_TEMPLATE(), AP\_QUEUE\_STD\_DEFN\_TEMPLATE(), barrier\_queue\_Delete(), MC\_RetrieveAgent(), MC\_WaitRetrieveAgent(), message\_queue\_SendOutgoing(), syncListDelete(), and syncListRemove().

#### 13.50.1.9 void ListTerminate (list\_p list)

Definition at line 49 of file list.c.

References list\_s::listhead, and list\_s::size.

Referenced by barrier\_queue\_Destroy(), df\_Destroy(), and df\_request\_list\_Destroy().

## 13.51 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mc\_list/list.h File Reference

```
#include <stdio.h>
#include <stdlib.h>
```

### Data Structures

- struct [listNode\\_s](#)
- struct [list\\_s](#)

### Defines

- #define [DATA](#) void\*
- #define [QUEUE\\_TEMPLATE](#)(name, node\_type, search\_type, search\_var\_name)

### Typedefs

- typedef struct [listNode\\_s](#) [listNode\\_t](#)
- typedef [listNode\\_t](#) \* [listNode\\_p](#)
- typedef struct [list\\_s](#) [list\\_t](#)
- typedef [list\\_t](#) \* [list\\_p](#)

### Functions

- [list\\_p](#) [ListInitialize](#) (void)
- void [ListTerminate](#) ([list\\_p](#) list)
- int [ListGetSize](#) ([list\\_p](#) list)
- int [ListAdd](#) ([list\\_p](#) list, [DATA](#) data)
- int [ListInsert](#) ([list\\_p](#) list, [DATA](#) data, const int index)
- [DATA](#) [ListGetHead](#) ([list\\_p](#) list)
- [DATA](#) [ListPop](#) ([list\\_p](#) list)
- [DATA](#) [ListSearch](#) ([list\\_p](#) list, const int index)
- [DATA](#) [ListDelete](#) ([list\\_p](#) list, const int index)

#### 13.51.1 Define Documentation

##### 13.51.1.1 #define DATA void\*

Definition at line 30 of file list.h.

Referenced by [barrier\\_queue\\_Add\(\)](#), [ListDelete\(\)](#), [ListPop\(\)](#), and [syncListAddNode\(\)](#).

##### 13.51.1.2 #define QUEUE\_TEMPLATE(name, node\_type, search\_type, search\_var\_name)

Definition at line 68 of file list.h.

## 13.51.2 Typedef Documentation

### 13.51.2.1 typedef list\_t\* list\_p

Definition at line 49 of file list.h.

### 13.51.2.2 typedef struct list\_s list\_t

### 13.51.2.3 typedef listNode\_t\* listNode\_p

Definition at line 39 of file list.h.

### 13.51.2.4 typedef struct listNode\_s listNode\_t

## 13.51.3 Function Documentation

### 13.51.3.1 int ListAdd (list\_p *list*, DATA *data*)

Definition at line 161 of file list.c.

References list\_s::listhead, listNode\_s::next, listNode\_s::node\_data, and list\_s::size.

Referenced by agent\_task\_Copy(), barrier\_queue\_Add(), df\_Add(), df\_AddRequest(), and syncListAddNode().

### 13.51.3.2 DATA ListDelete (list\_p *list*, const int *index*)

Definition at line 216 of file list.c.

References DATA, list\_s::listhead, listNode\_s::next, listNode\_s::node\_data, and list\_s::size.

Referenced by barrier\_queue\_Delete(), syncListDelete(), and syncListRemove().

### 13.51.3.3 DATA ListGetHead (list\_p *list*)

Definition at line 79 of file list.c.

References list\_s::listhead, and listNode\_s::node\_data.

### 13.51.3.4 int ListGetSize (list\_p *list*)

### 13.51.3.5 list\_p ListInitialize (void)

Definition at line 33 of file list.c.

References list\_s::listhead, and list\_s::size.

Referenced by barrier\_queue\_New(), df\_Initialize(), df\_request\_list\_New(), and syncListInit().

### 13.51.3.6 int ListInsert (list\_p *list*, DATA *data*, const int *index*)

Definition at line 203 of file list.c.

### 13.51.3.7 DATA ListPop (*list\_p list*)

Definition at line 89 of file list.c.

References DATA, list\_s::listhead, listNode\_s::next, listNode\_s::node\_data, and list\_s::size.

Referenced by AP\_QUEUE\_STD\_DEFN\_TEMPLATE(), barrier\_queue\_Pop(), df\_Destroy(), df\_request\_list\_Destroy(), and df\_request\_list\_Pop().

### 13.51.3.8 DATA ListSearch (*list\_p list*, *const int index*)

Definition at line 124 of file list.c.

References list\_s::listhead, listNode\_s::next, listNode\_s::node\_data, and list\_s::size.

Referenced by agent\_queue\_Flush(), agent\_task\_Copy(), ams\_ManageAgentList(), ams\_Print(), AP\_QUEUE\_SEARCH\_TEMPLATE(), AP\_QUEUE\_STD\_DEFN\_TEMPLATE(), barrier\_queue\_Delete(), MC\_RetrieveAgent(), MC\_WaitRetrieveAgent(), message\_queue\_SendOutgoing(), syncListDelete(), and syncListRemove().

### 13.51.3.9 void ListTerminate (*list\_p list*)

Definition at line 49 of file list.c.

References list\_s::listhead, and list\_s::size.

Referenced by barrier\_queue\_Destroy(), df\_Destroy(), and df\_request\_list\_Destroy().

## 13.52 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mc\_platform.c File Reference

```
#include "config.h"
#include <unistd.h>
#include <sys/socket.h>
#include <netdb.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include "include/acc.h"
#include "include/mc_platform.h"
#include "include/macros.h"
```

### Defines

- `#define` [DEFAULT\\_HOSTNAME\\_LENGTH](#) 200

### Functions

- [mc\\_platform\\_p mc\\_platform\\_Initialize](#) (MCAgency\_t agency, ChOptions\_t \*ch\_options)
- [int mc\\_platform\\_Destroy](#) (mc\_platform\_p platform)

### 13.52.1 Define Documentation

#### 13.52.1.1 `#define` [DEFAULT\\_HOSTNAME\\_LENGTH](#) 200

Definition at line 51 of file mc\_platform.c.

Referenced by [mc\\_platform\\_Initialize\(\)](#).

### 13.52.2 Function Documentation

#### 13.52.2.1 `int` [mc\\_platform\\_Destroy](#) (mc\_platform\_p *platform*)

Definition at line 244 of file mc\_platform.c.

References [mc\\_platform\\_s::acc](#), [acc\\_Destroy\(\)](#), [mc\\_platform\\_s::agent\\_queue](#), [mc\\_platform\\_s::ams](#), [ams\\_Destroy\(\)](#), [mc\\_platform\\_s::barrier\\_queue](#), [barrier\\_queue\\_Destroy\(\)](#), [mc\\_platform\\_s::cmd\\_prompt](#), [cmd\\_prompt\\_Destroy\(\)](#), [COND\\_DESTROY](#), [mc\\_platform\\_s::connection\\_queue](#), [mc\\_platform\\_s::df](#), [df\\_Destroy\(\)](#), [mc\\_platform\\_s::giant\\_cond](#), [mc\\_platform\\_s::giant\\_lock](#), [mc\\_platform\\_s::interp\\_options](#), [mc\\_platform\\_s::MC\\_signal\\_cond](#), [mc\\_platform\\_s::MC\\_signal\\_lock](#), [mc\\_platform\\_s::MC\\_steel\\_cond](#), [mc\\_platform\\_s::MC\\_steel\\_lock](#), [MC\\_SUCCESS](#), [mc\\_platform\\_s::MC\\_sync\\_cond](#), [mc\\_platform\\_s::MC\\_sync\\_lock](#), [mc\\_platform\\_s::message\\_queue](#), [MUTEX\\_DESTROY](#), [mc\\_platform\\_s::quit\\_lock](#), [SOCKET\\_ERROR](#), and [mc\\_platform\\_s::sockfd](#).

Referenced by [MC\\_End\(\)](#).

### 13.52.2.2 `mc_platform_p mc_platform_Initialize (MCAgency_t agency, ChOptions_t * ch_options)`

Definition at line 53 of file `mc_platform.c`.

References `acc_Initialize()`, `acc_Start()`, `agent_ChScriptInitVar()`, `ams_Initialize()`, `ams_Start()`, `barrier_queue_New()`, `CHECK_NULL`, `cmd_prompt_Initialize()`, `cmd_prompt_Start()`, `COND_INIT`, `COND_T`, `COND_WAIT`, `agency_s::default_agentstatus`, `DEFAULT_HOSTNAME_LENGTH`, `df_Initialize()`, `df_Start()`, `GET_THREAD_MODE`, `agency_s::initInterps`, `agency_s::last_error`, `MC_ERR_MEMORY`, `MC_NO_SIGNAL`, `mc_platform`, `MC_THREAD_ACC`, `MC_THREAD_ALL`, `MC_THREAD_AMS`, `MC_THREAD_CP`, `MC_THREAD_DF`, `MUTEX_INIT`, `MUTEX_LOCK`, `MUTEX_T`, `MUTEX_UNLOCK`, `agency_s::portno`, `agency_s::stack_size`, `syncListInit()`, and `agency_s::threads`.

Referenced by `MC_Initialize()`.

## 13.53 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mc\_rwlock.c File Reference

```
#include "config.h"
#include <stdio.h>
#include <stdlib.h>
#include "include/macros.h"
#include "include/mc_error.h"
#include "include/mc_rwlock.h"
```

### Functions

- [int mc\\_rwlock\\_init \(mc\\_rwlock\\_p rwlock\)](#)
- [int mc\\_rwlock\\_destroy \(mc\\_rwlock\\_p rwlock\)](#)
- [int mc\\_rwlock\\_rdlock \(mc\\_rwlock\\_p rwlock\)](#)
- [int mc\\_rwlock\\_rdunlock \(mc\\_rwlock\\_p rwlock\)](#)
- [int mc\\_rwlock\\_wrlock \(mc\\_rwlock\\_p rwlock\)](#)
- [int mc\\_rwlock\\_wrunlock \(mc\\_rwlock\\_p rwlock\)](#)

### 13.53.1 Function Documentation

#### 13.53.1.1 [int mc\\_rwlock\\_destroy \(mc\\_rwlock\\_p rwlock\)](#)

Definition at line 66 of file mc\_rwlock.c.

References [mc\\_rwlock\\_s::cond](#), [COND\\_DESTROY](#), [mc\\_rwlock\\_s::lock](#), and [MUTEX\\_DESTROY](#).

#### 13.53.1.2 [int mc\\_rwlock\\_init \(mc\\_rwlock\\_p rwlock\)](#)

Definition at line 46 of file mc\_rwlock.c.

References [CHECK\\_NULL](#), [mc\\_rwlock\\_s::cond](#), [COND\\_INIT](#), [COND\\_T](#), [mc\\_rwlock\\_s::lock](#), [MC\\_ERR\\_MEMORY](#), [MUTEX\\_INIT](#), [MUTEX\\_T](#), [mc\\_rwlock\\_s::num\\_readers](#), [mc\\_rwlock\\_s::write\\_flag](#), and [mc\\_rwlock\\_s::write\\_request](#).

#### 13.53.1.3 [int mc\\_rwlock\\_rdlock \(mc\\_rwlock\\_p rwlock\)](#)

Definition at line 79 of file mc\_rwlock.c.

References [mc\\_rwlock\\_s::cond](#), [COND\\_WAIT](#), [mc\\_rwlock\\_s::lock](#), [MUTEX\\_LOCK](#), [MUTEX\\_UNLOCK](#), [mc\\_rwlock\\_s::num\\_readers](#), [mc\\_rwlock\\_s::write\\_flag](#), and [mc\\_rwlock\\_s::write\\_request](#).

#### 13.53.1.4 [int mc\\_rwlock\\_rdunlock \(mc\\_rwlock\\_p rwlock\)](#)

Definition at line 94 of file mc\_rwlock.c.

References [mc\\_rwlock\\_s::cond](#), [COND\\_SIGNAL](#), [mc\\_rwlock\\_s::lock](#), [MUTEX\\_LOCK](#), [MUTEX\\_UNLOCK](#), and [mc\\_rwlock\\_s::num\\_readers](#).

**13.53.1.5 int mc\_rwlock\_wrlock (mc\_rwlock\_p *rwlock*)**

Definition at line 107 of file mc\_rwlock.c.

References mc\_rwlock\_s::cond, COND\_WAIT, mc\_rwlock\_s::lock, MUTEX\_LOCK, MUTEX\_UNLOCK, mc\_rwlock\_s::num\_readers, mc\_rwlock\_s::write\_flag, and mc\_rwlock\_s::write\_request.

**13.53.1.6 int mc\_rwlock\_wrunlock (mc\_rwlock\_p *rwlock*)**

Definition at line 126 of file mc\_rwlock.c.

References mc\_rwlock\_s::cond, COND\_SIGNAL, mc\_rwlock\_s::lock, MUTEX\_LOCK, MUTEX\_UNLOCK, and mc\_rwlock\_s::write\_flag.



## 13.54 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mc\_sync/sync\_list.c File Reference

```
#include <pthread.h>
#include "sync_list.h"
#include "../mc_list/list.h"
#include "../include/mc_error.h"
```

### Functions

- [int syncListNodeInit](#) (struct [syncListNode\\_s](#) \**node*)
- [struct syncListNode\\_s \\* syncListNodeNew](#) (void)
- [int syncListNodeDestroy](#) (struct [syncListNode\\_s](#) \**node*)
- [syncListNode\\_t \\* syncListFind](#) (int *id*, struct [syncList\\_s](#) \**list*)
- [struct syncList\\_s \\* syncListInit](#) (void)
- [int syncListAddNode](#) (struct [syncListNode\\_s](#) \**node*, struct [syncList\\_s](#) \**list*)
- [int syncListNew](#) (int *id*, struct [syncList\\_s](#) \**list*)
- [int syncListDelete](#) (int *id*, struct [syncList\\_s](#) \**list*)
- [syncListNode\\_t \\* syncListRemove](#) (int *id*, struct [syncList\\_s](#) \**list*)

### 13.54.1 Function Documentation

#### 13.54.1.1 [int syncListAddNode](#) (struct [syncListNode\\_s](#) \* *node*, struct [syncList\\_s](#) \* *list*)

Definition at line 86 of file [sync\\_list.c](#).

References [DATA](#), [syncListNode\\_s::id](#), [syncList\\_s::list](#), [ListAdd\(\)](#), [list\\_s::listhead](#), [syncList\\_s::lock](#), [listNode\\_s::next](#), [listNode\\_s::node\\_data](#), [RWLOCK\\_WRLock](#), and [RWLOCK\\_WRunlock](#).

Referenced by [MC\\_SyncInit\(\)](#), and [syncListNew\(\)](#).

#### 13.54.1.2 [int syncListDelete](#) (int *id*, struct [syncList\\_s](#) \* *list*)

Definition at line 115 of file [sync\\_list.c](#).

References [syncListNode\\_s::id](#), [syncList\\_s::list](#), [ListDelete\(\)](#), [ListSearch\(\)](#), [syncList\\_s::lock](#), [MC\\_ERR\\_NOT\\_FOUND](#), [RWLOCK\\_WRLock](#), [RWLOCK\\_WRunlock](#), [list\\_s::size](#), and [syncListNodeDestroy\(\)](#).

#### 13.54.1.3 [syncListNode\\_t\\* syncListFind](#) (int *id*, struct [syncList\\_s](#) \* *list*)

Definition at line 56 of file [sync\\_list.c](#).

References [syncList\\_s::list](#), [list\\_s::listhead](#), [syncList\\_s::lock](#), [listNode\\_s::next](#), [listNode\\_s::node\\_data](#), [RWLOCK\\_RDlock](#), and [RWLOCK\\_RDUnblock](#).

Referenced by [MC\\_CondBroadcast\(\)](#), [MC\\_CondReset\(\)](#), [MC\\_CondSignal\(\)](#), [MC\\_CondWait\(\)](#), [MC\\_MutexLock\(\)](#), [MC\\_MutexUnlock\(\)](#), [MC\\_SemaphorePost\(\)](#), [MC\\_SemaphoreWait\(\)](#), [MC\\_SyncDelete\(\)](#), and [MC\\_SyncInit\(\)](#).

**13.54.1.4 struct syncList\_s\* syncListInit (void) [read]**

Definition at line 72 of file sync\_list.c.

References ListInitialize(), syncList\_s::lock, MUTEX\_INIT, MUTEX\_T, RWLOCK\_INIT, and RWLOCK\_T.

Referenced by mc\_platform\_Initialize().

**13.54.1.5 int syncListNew (int id, struct syncList\_s \* list)**

Definition at line 105 of file sync\_list.c.

References node, syncListAddNode(), and syncListNodeInit().

**13.54.1.6 int syncListNodeDestroy (struct syncListNode\_s \* node)**

Definition at line 44 of file sync\_list.c.

References syncListNode\_s::cond, COND\_DESTROY, syncListNode\_s::lock, MUTEX\_DESTROY, syncListNode\_s::sem, and SEMAPHORE\_DESTROY.

Referenced by MC\_SyncDelete(), and syncListDelete().

**13.54.1.7 int syncListNodeInit (struct syncListNode\_s \* node)**

Definition at line 11 of file sync\_list.c.

References CHECK\_NULL, syncListNode\_s::cond, COND\_INIT, COND\_T, syncListNode\_s::lock, MUTEX\_INIT, MUTEX\_T, syncListNode\_s::sem, SEMAPHORE\_INIT, and SEMAPHORE\_T.

Referenced by syncListNew().

**13.54.1.8 struct syncListNode\_s\* syncListNodeNew (void) [read]**

Definition at line 26 of file sync\_list.c.

References CHECK\_NULL, syncListNode\_s::cond, COND\_INIT, COND\_T, syncListNode\_s::lock, MUTEX\_INIT, MUTEX\_T, syncListNode\_s::sem, SEMAPHORE\_INIT, SEMAPHORE\_T, and syncListNode\_s::signalled.

Referenced by MC\_SyncInit().

**13.54.1.9 syncListNode\_t\* syncListRemove (int id, struct syncList\_s \* list)**

Definition at line 132 of file sync\_list.c.

References syncListNode\_s::id, syncList\_s::list, ListDelete(), ListSearch(), syncList\_s::lock, RWLOCK\_WRLock, RWLOCK\_WRunLock, and list\_s::size.

Referenced by MC\_SyncDelete().

## 13.55 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mc\_sync/sync\_list.h File Reference

```
#include "../include/macros.h"
#include "../mc_list/list.h"
#include "../include/mc_rwlock.h"
```

### Data Structures

- struct [syncListNode\\_s](#)
- struct [syncList\\_s](#)

### Typedefs

- typedef struct [syncListNode\\_s](#) [syncListNode\\_t](#)
- typedef [syncListNode\\_t](#) \* [syncListNode\\_p](#)
- typedef struct [syncList\\_s](#) [syncList\\_t](#)
- typedef [syncList\\_t](#) \* [syncList\\_p](#)

### Functions

- [int syncListNodeInit](#) (struct [syncListNode\\_s](#) \*[node](#))
- [int syncListNodeDestroy](#) (struct [syncListNode\\_s](#) \*[node](#))
- [syncListNode\\_t](#) \* [syncListFind](#) (int id, struct [syncList\\_s](#) \*[list](#))
- struct [syncListNode\\_s](#) \* [syncListNodeNew](#) (void)
- [int syncListDelete](#) (int id, struct [syncList\\_s](#) \*[list](#))
- [syncListNode\\_t](#) \* [syncListRemove](#) (int id, struct [syncList\\_s](#) \*[list](#))
- struct [syncList\\_s](#) \* [syncListInit](#) (void)
- [int syncListAddNode](#) (struct [syncListNode\\_s](#) \*[node](#), struct [syncList\\_s](#) \*[list](#))
- [int syncListNew](#) (int id, struct [syncList\\_s](#) \*[list](#))
- [syncListNode\\_t](#) \* [syncListGet](#) (int id, struct [syncList\\_s](#) \*[list](#))

#### 13.55.1 Typedef Documentation

##### 13.55.1.1 typedef [syncList\\_t](#)\* [syncList\\_p](#)

Definition at line 33 of file [sync\\_list.h](#).

##### 13.55.1.2 typedef struct [syncList\\_s](#) [syncList\\_t](#)

##### 13.55.1.3 typedef [syncListNode\\_t](#)\* [syncListNode\\_p](#)

Definition at line 23 of file [sync\\_list.h](#).

### 13.55.1.4 typedef struct syncListNode\_s syncListNode\_t

## 13.55.2 Function Documentation

### 13.55.2.1 int syncListAddNode (struct syncListNode\_s \* *node*, struct syncList\_s \* *list*)

Definition at line 86 of file sync\_list.c.

References DATA, syncListNode\_s::id, syncList\_s::list, ListAdd(), list\_s::listhead, syncList\_s::lock, listNode\_s::next, listNode\_s::node\_data, RWLOCK\_WRLock, and RWLOCK\_WRunlock.

Referenced by MC\_SyncInit(), and syncListNew().

### 13.55.2.2 int syncListDelete (int *id*, struct syncList\_s \* *list*)

Definition at line 115 of file sync\_list.c.

References syncListNode\_s::id, syncList\_s::list, ListDelete(), ListSearch(), syncList\_s::lock, MC\_ERR\_NOT\_FOUND, RWLOCK\_WRLock, RWLOCK\_WRunlock, list\_s::size, and syncListNodeDestroy().

### 13.55.2.3 syncListNode\_t\* syncListFind (int *id*, struct syncList\_s \* *list*)

Definition at line 56 of file sync\_list.c.

References syncList\_s::list, list\_s::listhead, syncList\_s::lock, listNode\_s::next, listNode\_s::node\_data, RWLOCK\_RDLOCK, and RWLOCK\_RDUnlock.

Referenced by MC\_CondBroadcast(), MC\_CondReset(), MC\_CondSignal(), MC\_CondWait(), MC\_MutexLock(), MC\_MutexUnlock(), MC\_SemaphorePost(), MC\_SemaphoreWait(), MC\_SyncDelete(), and MC\_SyncInit().

### 13.55.2.4 syncListNode\_t\* syncListGet (int *id*, struct syncList\_s \* *list*)

### 13.55.2.5 struct syncList\_s\* syncListInit (void) [read]

Definition at line 72 of file sync\_list.c.

References ListInitialize(), syncList\_s::lock, MUTEX\_INIT, MUTEX\_T, RWLOCK\_INIT, and RWLOCK\_T.

Referenced by mc\_platform\_Initialize().

### 13.55.2.6 int syncListNew (int *id*, struct syncList\_s \* *list*)

Definition at line 105 of file sync\_list.c.

References node, syncListAddNode(), and syncListNodeInit().

### 13.55.2.7 int syncListNodeDestroy (struct syncListNode\_s \* *node*)

Definition at line 44 of file sync\_list.c.

References syncListNode\_s::cond, COND\_DESTROY, syncListNode\_s::lock, MUTEX\_DESTROY, syncListNode\_s::sem, and SEMAPHORE\_DESTROY.

Referenced by MC\_SyncDelete(), and syncListDelete().

#### **13.55.2.8 int syncListNodeInit (struct syncListNode\_s \* *node*)**

Definition at line 11 of file sync\_list.c.

References CHECK\_NULL, syncListNode\_s::cond, COND\_INIT, COND\_T, syncListNode\_s::lock, MUTEX\_INIT, MUTEX\_T, syncListNode\_s::sem, SEMAPHORE\_INIT, and SEMAPHORE\_T.

Referenced by syncListNew().

#### **13.55.2.9 struct syncListNode\_s\* syncListNodeNew (void) [read]**

Definition at line 26 of file sync\_list.c.

References CHECK\_NULL, syncListNode\_s::cond, COND\_INIT, COND\_T, syncListNode\_s::lock, MUTEX\_INIT, MUTEX\_T, syncListNode\_s::sem, SEMAPHORE\_INIT, SEMAPHORE\_T, and syncListNode\_s::signalled.

Referenced by MC\_SyncInit().

#### **13.55.2.10 syncListNode\_t\* syncListRemove (int *id*, struct syncList\_s \* *list*)**

Definition at line 132 of file sync\_list.c.

References syncListNode\_s::id, syncList\_s::list, ListDelete(), ListSearch(), syncList\_s::lock, RWLOCK\_WRLOCK, RWLOCK\_WRUNLOCK, and list\_s::size.

Referenced by MC\_SyncDelete().

## 13.56 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/message.c File Reference

```
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <unistd.h>
#include <netdb.h>
#include <errno.h>
#include "config.h"
#include <mxml.h>
#include "include/libmc.h"
#include "include/agent.h"
#include "include/mc_platform.h"
#include "include/message.h"
#include "include/mtp_http.h"
#include "include/xml_compose.h"
#include "include/xml_helper.h"
#include "include/xml_parser.h"
#include "include/macros.h"
#include "security/asm_node.h"
```

### Defines

- `#define SOCKET_INPUT_SIZE 4096`
- `#define MSG_THREADS 40`
- `#define MSG_THREAD_EXIT()`

### Functions

- `message_p message_New (void)`
- `message_p message_Copy (message_p src)`
- `int message_InitializeFromAgent (mc_platform_p mc_platform, message_p message, agent_p agent)`
- `int message_InitializeFromConnection (mc_platform_p mc_platform, message_p message, connection_p connection)`
- `int http_to_hostport (const char *http_str, char **host, int *port, char **target)`
- `int message_InitializeFromString (mc_platform_p mc_platform, message_p message, const char *string, const char *destination_host, int destination_port, const char *target)`
- `int message_Destroy (message_p message)`

- `int auth_rece_send_msg` (`int sockfd`, `char *hostname`, `char *message`, `char *privkey`, `char *known_host_filename`)
- `int message_Send` (`mc_platform_t *mc_platform`, `message_p message`, `char *privatekey`)
- `void * message_send_Thread` (`void *arg`)

## 13.56.1 Define Documentation

### 13.56.1.1 #define MSG\_THREAD\_EXIT()

Value:

```
free(arg); \
message_Destroy(message); \
MUTEX_LOCK(&mc_platform->acc->msg_thread_lock); \
mc_platform->acc->num_msg_threads--; \
COND_SIGNAL(&mc_platform->acc->msg_thread_cond); \
MUTEX_UNLOCK(&mc_platform->acc->msg_thread_lock); \
THREAD_EXIT()
```

Definition at line 590 of file message.c.

Referenced by `message_send_Thread()`.

### 13.56.1.2 #define MSG\_THREADS 40

Definition at line 561 of file message.c.

Referenced by `message_Send()`.

### 13.56.1.3 #define SOCKET\_INPUT\_SIZE 4096

Definition at line 61 of file message.c.

## 13.56.2 Function Documentation

### 13.56.2.1 int auth\_rece\_send\_msg(*int sockfd, char \* hostname, char \* message, char \* privkey, char \* known\_host\_filename*)

Definition at line 437 of file message.c.

References `aes_en_de()`, `initiate_migration_process()`, `read_known_host_file()`, and `send_AES_en_MA()`.

Referenced by `message_send_Thread()`.

### 13.56.2.2 int http\_to\_hostport(*const char \* http\_str, char \*\* host, int \* port, char \*\* target*)

Definition at line 287 of file message.c.

References `MC_ERR_PARSE`.

Referenced by `MC_AclSend()`.

### 13.56.2.3 `message_p message_Copy (message_p src)`

Definition at line 88 of file `message.c`.

### 13.56.2.4 `int message_Destroy (message_p message)`

Definition at line 398 of file `message.c`.

References `message_s::addr`, `message_s::agent_xml_flag`, `message_s::from_address`, `MC_SUCCESS`, `message_s::message_body`, `mxmlDelete()`, `message_s::target`, `message_s::to_address`, `message_s::update_name`, and `message_s::xml_root`.

Referenced by `acc_connection_Thread()`, `acc_MessageHandlerThread()`, `ams_ManageAgentList()`, `MC_LoadAgentFromFile()`, `MC_SendAgentMigrationMessage()`, `MC_SendAgentMigrationMessageFile()`, `message_InitializeFromConnection()`, and `message_InitializeFromString()`.

### 13.56.2.5 `int message_InitializeFromAgent (mc_platform_p mc_platform, message_p message, agent_p agent)`

Definition at line 96 of file `message.c`.

References `message_s::addr`, `agent_xml_compose()`, `message_s::agent_xml_flag`, `buf`, `CHECK_NULL`, `agent_s::datastate`, `mc_platform_s::err`, `message_s::from_address`, `agent_s::home`, `mc_platform_s::hostname`, `MC_ERR_MEMORY`, `MC_SUCCESS`, `message_s::message_body`, `message_s::message_id`, `message_s::message_type`, `MOBILE_AGENT`, `MXML_NO_CALLBACK`, `mxmlSaveAllocString()`, `agent_s::name`, `agent_datastate_s::number_of_tasks`, `mc_platform_s::port`, `message_s::sending_agent_name`, `agent_task_s::server_name`, `strtok_r`, `message_s::target`, `agent_datastate_s::task_progress`, `agent_datastate_s::tasks`, `message_s::to_address`, `message_s::update_name`, `WARN`, and `message_s::xml_root`.

### 13.56.2.6 `int message_InitializeFromConnection (mc_platform_p mc_platform, message_p message, connection_p connection)`

Definition at line 196 of file `message.c`.

References `connection_s::addr`, `message_s::addr`, `CHECK_NULL`, `connection_s::clientfd`, `connection_s::connect_id`, `message_s::connect_id`, `message_s::from_address`, `MC_ERR_CONNECT`, `MC_ERR_PARSE`, `MC_SUCCESS`, `message_s::message_body`, `message_Destroy()`, `message_s::message_id`, `message_xml_parse()`, `MXML_NO_CALLBACK`, `mxmlLoadString()`, `SOCKET_ERROR`, `SOCKET_INPUT_SIZE`, `message_s::target`, `message_s::to_address`, and `message_s::xml_root`.

### 13.56.2.7 `int message_InitializeFromString (mc_platform_p mc_platform, message_p message, const char * string, const char * destination_host, int destination_port, const char * target)`

Definition at line 322 of file `message.c`.

References `message_s::addr`, `CHECK_NULL`, `message_s::connect_id`, `mc_platform_s::err`, `message_s::from_address`, `mc_platform_s::hostname`, `MC_ERR_MEMORY`, `MC_ERR_PARSE`, `MC_SUCCESS`, `message_s::message_body`, `message_Destroy()`, `message_s::message_id`, `message_s::message_type`, `message_xml_parse()`, `MOBILE_AGENT`, `MXML_NO_CALLBACK`, `mxmlLoadString()`, `mc_platform_s::port`, `message_s::target`, `message_s::to_address`, `message_s::update_name`, and `message_s::xml_root`.



### 13.56.2.8 `message_p message_New (void)`

Definition at line 64 of file message.c.

References `message_s::addr`, `message_s::agent_xml_flag`, `CHECK_NULL`, `message_s::connect_id`, `message_s::from_address`, `message_s::http_type`, `message_s::isHTTP`, `message_s::message_body`, `message_s::message_id`, `message_s::message_type`, `message_s::target`, `message_s::to_address`, `message_s::update_name`, `message_s::update_num`, `message_s::xml_payload`, and `message_s::xml_root`.

Referenced by `acc_connection_Thread()`, `ams_ManageAgentList()`, `MC_LoadAgentFromFile()`, `MC_SendAgentMigrationMessage()`, `MC_SendAgentMigrationMessageFile()`, and `mtp_http_CreateMessage()`.

### 13.56.2.9 `int message_Send (mc_platform_t * mc_platform, message_p message, char * privatekey)`

Definition at line 562 of file message.c.

References `mc_platform_s::acc`, `COND_WAIT`, `message_send_arg_s::mc_platform`, `message_send_arg_s::message`, `message_send_Thread()`, `MSG_THREADS`, `MUTEX_LOCK`, `MUTEX_UNLOCK`, `message_send_arg_s::privatekey`, `THREAD_CREATE`, `THREAD_DETACH`, and `THREAD_T`.

Referenced by `acc_MessageHandlerThread()`, `MC_AclSend()`, and `message_queue_SendOutgoing()`.

### 13.56.2.10 `void* message_send_Thread (void * arg)`

Definition at line 601 of file message.c.

References `mc_platform_s::agency`, `auth_rece_send_msg()`, `buf`, `CHECK_NULL`, `dynstring_Append()`, `dynstring_Destroy()`, `dynstring_New()`, `agency_s::known_host_filename`, `mc_platform`, `dynstring_s::message`, `message_s::message_body`, `MSG_THREAD_EXIT`, `mtp_http_ComposeMessage()`, `mtp_http_Destroy()`, `mtp_http_New()`, `mtp_http_Parse()`, `port`, `send`, `message_s::sending_agent_name`, `SOCKET_ERROR`, `SOCKET_INPUT_SIZE`, `strtok_r`, and `message_s::to_address`.

Referenced by `message_Send()`.

## 13.57 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/message\_queue.c File Reference

```
#include "config.h"
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include "include/data_structures.h"
#include "include/mc_platform.h"
```

### Functions

- void [message\\_queue\\_SendOutgoing](#) (struct [mc\\_platform\\_s](#) \*[mc\\_platform](#), [message\\_queue\\_p](#) [mqueue](#))

#### 13.57.1 Function Documentation

##### 13.57.1.1 void [message\\_queue\\_SendOutgoing](#) (struct [mc\\_platform\\_s](#) \* [mc\\_platform](#), [message\\_queue\\_p](#) [mqueue](#))

Definition at line 57 of file [message\\_queue.c](#).

References [CHECK\\_NULL](#), [mc\\_platform\\_s::hostname](#), [ListSearch\(\)](#), [message\\_Send\(\)](#), [MUTEX\\_LOCK](#), [MUTEX\\_UNLOCK](#), [mc\\_platform\\_s::port](#), and [message\\_s::to\\_address](#).

## 13.58 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mtp\_http.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <unistd.h>
#include <errno.h>
#include "config.h"
#include "include/connection.h"
#include "include/mtp_http.h"
#include "include/macros.h"
#include "include/mc_error.h"
#include "include/message.h"
#include "include/dynstring.h"
```

### Defines

- #define [SAFE\\_FREE](#)(elem)
- #define [HTTP\\_PARSE\\_EXPR](#)(parse\_name, struct\_name)
- #define [SAFE\\_FREE](#)(object)

### Functions

- [int mtp\\_http\\_Destroy](#) ([mtp\\_http\\_p](#) http)
- [mtp\\_http\\_p mtp\\_http\\_New](#) (void)
- [int rece\\_de\\_msg](#) (char \*buffer, [connection\\_p](#) con, char \*privatekey)
- [int mtp\\_http\\_InitializeFromConnection](#) ([mtp\\_http\\_p](#) http, [connection\\_p](#) connection, char \*privatekey)
- const char \* [http\\_GetExpression](#) (const char \*string, char \*\*expr)

*Parse an html expression.*

- [int http\\_ParseExpression](#) (const char \*expression\_string, char \*\*name, char \*\*value)

*Parse an expression into its name and value.*

- const char \* [mtp\\_http\\_ParseHeader](#) (struct [mtp\\_http\\_s](#) \*http, const char \*string)
- [int mtp\\_http\\_Parse](#) (struct [mtp\\_http\\_s](#) \*http, const char \*string)
- const char \* [http\\_ParseRequest](#) ([mtp\\_http\\_p](#) http, const char \*string)
- const char \* [http\\_GetToken](#) (const char \*string, char \*\*token)
- [int mtp\\_http\\_ParseResponse](#) (struct [mtp\\_http\\_s](#) \*http, const char \*string)
- [int mtp\\_http\\_ComposeMessage](#) ([message\\_p](#) message)
- struct [message\\_s](#) \* [mtp\\_http\\_CreateMessage](#) ([mtp\\_http\\_t](#) \*mtp\_http, char \*hostname, [int](#) port)

### 13.58.1 Define Documentation

#### 13.58.1.1 #define HTTP\_PARSE\_EXPR(parse\_name, struct\_name)

**Value:**

```
if ( !strcmp(name, parse_name) ) { \
    http->struct_name = (char*)malloc \
    ( \
        sizeof(char) * \
        (strlen(value)+1) \
    ); \
    strcpy(http->struct_name, value); \
} else
```

Referenced by mtp\_http\_ParseHeader().

#### 13.58.1.2 #define SAFE\_FREE(object)

**Value:**

```
if(object) free(object); \
    object = NULL
```

#### 13.58.1.3 #define SAFE\_FREE(elem)

**Value:**

```
if(elem) \
    free(elem)
```

Referenced by mtp\_http\_Destroy(), and mtp\_http\_ParseHeader().

### 13.58.2 Function Documentation

#### 13.58.2.1 const char\* http\_GetExpression (const char \* *string*, char \*\* *expr*)

Parse an html expression.

**Parameters:**

*string* (input) The html block of text: Will parse the first expression pointed to by 'string'.

*expr* (output) The allocated expression

**Returns:**

A pointer to the next expression segment of the string block, or NULL.

Definition at line 347 of file mtp\_http.c.

Referenced by mtp\_http\_Parse(), and mtp\_http\_ParseHeader().

**13.58.2.2 const char\* http\_GetToken (const char \* *string*, char \*\* *token*)**

Definition at line 788 of file mtp\_http.c.

References `cur`.

Referenced by `http_ParseRequest()`.

**13.58.2.3 int http\_ParseExpression (const char \* *expression\_string*, char \*\* *name*, char \*\* *value*)**

Parse an expression into its name and value.

**Parameters:**

*expression\_string* (input) The expression

*name* (output) An allocated name string or NULL

*value* (output) An allocated value string or NULL

**Returns:**

`error_code_t` type

**Note:**

an http expression is something like 'Date: Mon, 23 May 2005 22:38:34 GMT  
' where 'Date' is the name and the remainder of the string is the value

Definition at line 406 of file mtp\_http.c.

References `CHECK_NULL`, `MC_ERR_PARSE`, and `MC_SUCCESS`.

Referenced by `mtp_http_Parse()`, and `mtp_http_ParseHeader()`.

**13.58.2.4 const char\* http\_ParseRequest (mtp\_http\_p *http*, const char \* *string*)**

Definition at line 699 of file mtp\_http.c.

References `cur`, `HTTP_CONNECT`, `HTTP_DELETE`, `HTTP_GET`, `http_GetToken()`, `HTTP_HEAD`, `HTTP_OPTIONS`, `mtp_http_s::http_performative`, `HTTP_PERFORMATIVE_UNDEF`, `HTTP_POST`, `HTTP_PUT`, `HTTP_RESPONSE`, `HTTP_TRACE`, `mtp_http_s::response_code`, `mtp_http_s::response_string`, and `mtp_http_s::target`.

Referenced by `mtp_http_ParseHeader()`.

**13.58.2.5 int mtp\_http\_ComposeMessage (message\_p *message*)**

Definition at line 827 of file mtp\_http.c.

References `buf`, `message_s::isHTTP`, `MC_SUCCESS`, `message_s::message_body`, `PACKAGE_VERSION`, `message_s::target`, and `message_s::to_address`.

**13.58.2.6 struct message\_s\* mtp\_http\_CreateMessage (mtp\_http\_t \* *mtp\_http*, char \* *hostname*, int *port*) [read]**

Definition at line 873 of file mtp\_http.c.

References `buf`, `mtp_http_s::content`, `mtp_http_content_s::content_type`, `mtp_http_content_s::data`, `dynstring_Append()`, `dynstring_Destroy()`, `dynstring_New()`, `mtp_http_s::host`, `message_s::isHTTP`, `dynstring_s::len`, `dynstring_s::message`, `message_s::message_body`, `message_New()`, `mtp_http_s::message_parts`, `PACKAGE_VERSION`, `mtp_http_s::target`, and `message_s::to_address`.

Referenced by `MC_AclSend()`.

### 13.58.2.7 `int mtp_http_Destroy (mtp_http_p http)`

Definition at line 54 of file `mtp_http.c`.

References `mtp_http_s::accept_ranges`, `mtp_http_s::boundary`, `mtp_http_s::connection`, `mtp_http_s::content`, `mtp_http_s::content_length`, `mtp_http_content_s::content_type`, `mtp_http_s::content_type`, `mtp_http_content_s::data`, `mtp_http_s::date`, `mtp_http_s::host`, `mtp_http_s::http_version`, `mtp_http_s::message_parts`, `mtp_http_s::response_string`, `mtp_http_s::return_code`, `SAFE_FREE`, `mtp_http_s::server`, `mtp_http_s::target`, and `mtp_http_s::user_agent`.

Referenced by `acc_connection_Thread()`, `MC_AclSend()`, `message_send_Thread()`, and `mtp_http_InitializeFromConnection()`.

### 13.58.2.8 `int mtp_http_InitializeFromConnection (mtp_http_p http, connection_p connection, char * privatekey)`

Definition at line 178 of file `mtp_http.c`.

References `CHECK_NULL`, `connection_s::clientfd`, `mtp_http_s::content_length`, `dynstring_Append()`, `dynstring_Destroy()`, `dynstring_New()`, `ERR`, `mtp_http_s::header_length`, `HTTP_HEAD`, `mtp_http_s::http_performative`, `MC_ERR_CONNECT`, `mtp_http_Destroy()`, `mtp_http_New()`, `mtp_http_Parse()`, `mtp_http_ParseHeader()`, `PACKAGE_STRING`, `rece_de_msg()`, `send`, `SOCKET_ERROR`, and `SOCKET_INPUT_SIZE`.

### 13.58.2.9 `mtp_http_p mtp_http_New (void)`

Definition at line 87 of file `mtp_http.c`.

References `CHECK_NULL`, and `mtp_http_s::content`.

Referenced by `acc_connection_Thread()`, `MC_AclSend()`, `message_send_Thread()`, and `mtp_http_InitializeFromConnection()`.

### 13.58.2.10 `int mtp_http_Parse (struct mtp_http_s * http, const char * string)`

Definition at line 549 of file `mtp_http.c`.

References `mtp_http_s::boundary`, `mtp_http_s::content`, `mtp_http_s::content_length`, `mtp_http_content_s::content_type`, `mtp_http_s::content_type`, `mtp_http_content_s::data`, `http_GetExpression()`, `HTTP_HEAD`, `http_ParseExpression()`, `mtp_http_s::http_performative`, `HTTP_POST`, `HTTP_PUT`, `HTTP_RESPONSE`, `MC_SUCCESS`, `mtp_http_s::message_parts`, and `mtp_http_ParseHeader()`.

Referenced by `message_send_Thread()`, and `mtp_http_InitializeFromConnection()`.

### 13.58.2.11 `const char* mtp_http_ParseHeader (struct mtp_http_s * http, const char * string)`

Definition at line 465 of file `mtp_http.c`.

References `mtp_http_s::header_length`, `http_GetExpression()`, `HTTP_PARSE_EXPR`, `http_ParseExpression()`, `http_ParseRequest()`, `MC_SUCCESS`, and `SAFE_FREE`.

Referenced by `mtp_http_InitializeFromConnection()`, and `mtp_http_Parse()`.

#### 13.58.2.12 `int mtp_http_ParseResponse (struct mtp_http_s * http, const char * string)`

Definition at line 821 of file `mtp_http.c`.

#### 13.58.2.13 `int rece_de_msg (char * buffer, connection_p con, char * privatekey)`

Definition at line 98 of file `mtp_http.c`.

References `aes_en_de()`, `connection_s::AES_key`, `connection_s::clientfd`, `int`, `connection_s::nonce`, and `receive_AES_en_MA()`.

Referenced by `mtp_http_InitializeFromConnection()`.

## 13.59 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mxml-2.2.2/config.h File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdarg.h>
#include <ctype.h>
```

### Defines

- `#define MXML_VERSION "Mini-XML v2.2.2"`
- `#define HAVE_VSNPRINTF 1`
- `#define HAVE_STRDUP 1`

### Functions

- `char * mxml_strdup (const char *, va_list)`

#### 13.59.1 Define Documentation

##### 13.59.1.1 `#define HAVE_STRDUP 1`

Definition at line 49 of file config.h.

##### 13.59.1.2 `#define HAVE_VSNPRINTF 1`

Definition at line 42 of file config.h.

##### 13.59.1.3 `#define MXML_VERSION "Mini-XML v2.2.2"`

Definition at line 35 of file config.h.

Referenced by `write_documentation()`.

#### 13.59.2 Function Documentation

##### 13.59.2.1 `char* mxml_strdup (const char *, va_list)`

Definition at line 62 of file mxml-string.c.

Referenced by `mxml_error()`, `mxmlNewTextf()`, and `mxmlSetTextf()`.



## 13.60 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mxml-2.2.2/vcnet/config.h File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdarg.h>
#include <ctype.h>
```

### Defines

- `#define` [MXML\\_VERSION](#) "Mini-XML v2.0"
- `#define` [HAVE\\_STRDUP](#) 1

### Functions

- `char *` [mxml\\_strdup](#) (`const char *`, `va_list`)

#### 13.60.1 Define Documentation

##### 13.60.1.1 `#define HAVE_STRDUP 1`

Definition at line 51 of file config.h.

##### 13.60.1.2 `#define MXML_VERSION "Mini-XML v2.0"`

Definition at line 37 of file config.h.

#### 13.60.2 Function Documentation

##### 13.60.2.1 `char* mxml_strdup (const char *, va_list)`

Definition at line 62 of file mxml-string.c.

## 13.61 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mxml-2.2.2/vcnet2005/config.h File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdarg.h>
#include <ctype.h>
```

### Defines

- `#define MXML_VERSION "Mini-XML v2.0"`
- `#define HAVE_STRDUP 1`

### Functions

- `char * mxml_strdup (const char *, va_list)`

#### 13.61.1 Define Documentation

##### 13.61.1.1 `#define HAVE_STRDUP 1`

Definition at line 51 of file config.h.

##### 13.61.1.2 `#define MXML_VERSION "Mini-XML v2.0"`

Definition at line 37 of file config.h.

#### 13.61.2 Function Documentation

##### 13.61.2.1 `char* mxml_strdup (const char *, va_list)`

Definition at line 62 of file mxml-string.c.

## 13.62 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mxml-2.2.2/vcnet2008/config.h File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdarg.h>
#include <ctype.h>
```

### Defines

- `#define` [MXML\\_VERSION](#) "Mini-XML v2.0"
- `#define` [HAVE\\_STRDUP](#) 1

### Functions

- `char *` [mxml\\_strdup](#) (`const char *`, `va_list`)

#### 13.62.1 Define Documentation

##### 13.62.1.1 `#define HAVE_STRDUP 1`

Definition at line 51 of file config.h.

##### 13.62.1.2 `#define MXML_VERSION "Mini-XML v2.0"`

Definition at line 37 of file config.h.

#### 13.62.2 Function Documentation

##### 13.62.2.1 `char* mxml_strdup (const char *, va_list)`

Definition at line 62 of file mxml-string.c.

References [mxml\\_vsnprintf\(\)](#).

## 13.63 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/config.h File Reference

### Defines

- `#define _CRT_SECURE_NO_DEPRECATED 1`
- `#define XYSSL_HAVE_ASM`
- `#define XYSSL_DEBUG_MSG`
- `#define XYSSL_SELF_TEST`
- `#define XYSSL_GENPRIME`
- `#define XYSSL_AES_C`
- `#define XYSSL_ARC4_C`
- `#define XYSSL_BASE64_C`
- `#define XYSSL_BIGNUM_C`
- `#define XYSSL_CERTS_C`
- `#define XYSSL_DEBUG_C`
- `#define XYSSL_DES_C`
- `#define XYSSL_DHM_C`
- `#define XYSSL_HAVEGE_C`
- `#define XYSSL_MD5_C`
- `#define XYSSL_NET_C`
- `#define XYSSL_PADLOCK_C`
- `#define XYSSL_RSA_C`
- `#define XYSSL_SHA1_C`
- `#define XYSSL_SHA2_C`
- `#define XYSSL_SHA4_C`
- `#define XYSSL_SSL_CLI_C`
- `#define XYSSL_SSL_SRV_C`
- `#define XYSSL_SSL_TLS_C`
- `#define XYSSL_TIMING_C`
- `#define XYSSL_X509_PARSE_C`
- `#define XYSSL_X509_WRITE_C`

### 13.63.1 Define Documentation

#### 13.63.1.1 `#define _CRT_SECURE_NO_DEPRECATED 1`

Definition at line 12 of file config.h.

#### 13.63.1.2 `#define XYSSL_AES_C`

Definition at line 74 of file config.h.

#### 13.63.1.3 `#define XYSSL_ARC4_C`

Definition at line 84 of file config.h.

### 13.63

/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/config.h File Reference 577

#### 13.63.1.4 **#define XYSSL\_BASE64\_C**

Definition at line 92 of file config.h.

#### 13.63.1.5 **#define XYSSL\_BIGNUM\_C**

Definition at line 103 of file config.h.

#### 13.63.1.6 **#define XYSSL\_CERTS\_C**

Definition at line 111 of file config.h.

#### 13.63.1.7 **#define XYSSL\_DEBUG\_C**

Definition at line 121 of file config.h.

#### 13.63.1.8 **#define XYSSL\_DEBUG\_MSG**

Definition at line 47 of file config.h.

#### 13.63.1.9 **#define XYSSL\_DES\_C**

Definition at line 131 of file config.h.

#### 13.63.1.10 **#define XYSSL\_DHM\_C**

Definition at line 142 of file config.h.

#### 13.63.1.11 **#define XYSSL\_GENPRIME**

Definition at line 57 of file config.h.

#### 13.63.1.12 **#define XYSSL\_HAVE\_ASM**

Definition at line 36 of file config.h.

#### 13.63.1.13 **#define XYSSL\_HAVEGE\_C**

Definition at line 150 of file config.h.

#### 13.63.1.14 **#define XYSSL\_MD5\_C**

Definition at line 177 of file config.h.

**13.63.1.15 #define XYSSL\_NET\_C**

Definition at line 185 of file config.h.

**13.63.1.16 #define XYSSL\_PADLOCK\_C**

Definition at line 193 of file config.h.

**13.63.1.17 #define XYSSL\_RSA\_C**

Definition at line 204 of file config.h.

**13.63.1.18 #define XYSSL\_SELF\_TEST**

Definition at line 52 of file config.h.

**13.63.1.19 #define XYSSL\_SHA1\_C**

Definition at line 215 of file config.h.

**13.63.1.20 #define XYSSL\_SHA2\_C**

Definition at line 223 of file config.h.

**13.63.1.21 #define XYSSL\_SHA4\_C**

Definition at line 231 of file config.h.

**13.63.1.22 #define XYSSL\_SSL\_CLI\_C**

Definition at line 239 of file config.h.

**13.63.1.23 #define XYSSL\_SSL\_SRV\_C**

Definition at line 247 of file config.h.

**13.63.1.24 #define XYSSL\_SSL\_TLS\_C**

Definition at line 256 of file config.h.

**13.63.1.25 #define XYSSL\_TIMING\_C**

Definition at line 264 of file config.h.

### 13.63

/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/config.h File Reference 579

**13.63.1.26** ~~#define XYSSL\_X509\_PARSE\_C~~

---

Definition at line 274 of file config.h.

**13.63.1.27** ~~#define XYSSL\_X509\_WRITE\_C~~

Definition at line 282 of file config.h.

## 13.64 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mxml-2.2.2/mxml-attr.c File Reference

```
#include "config.h"
#include "mxml.h"
```

### Functions

- const char \* [mxmlElementGetAttr](#) ([mxml\\_node\\_t](#) \**node*, const char \**name*)
- void [mxmlElementSetAttr](#) ([mxml\\_node\\_t](#) \**node*, const char \**name*, const char \**value*)

### 13.64.1 Function Documentation

#### 13.64.1.1 const char\* [mxmlElementGetAttr](#) ([mxml\\_node\\_t](#) \* *node*, const char \* *name*)

Definition at line 43 of file mxml-attr.c.

References [mxml\\_value\\_s::attrs](#), [mxml\\_value\\_u::element](#), [MXML\\_ELEMENT](#), [mxml\\_attr\\_s::name](#), [mxml\\_value\\_s::num\\_attrs](#), [mxml\\_node\\_s::type](#), [mxml\\_attr\\_s::value](#), and [mxml\\_node\\_s::value](#).

Referenced by [agent\\_xml\\_parse\\_\\_agent\\_code\(\)](#), [agent\\_xml\\_parse\\_\\_data\(\)](#), [agent\\_xml\\_parse\\_\\_task\(\)](#), [agent\\_xml\\_parse\\_\\_tasks\(\)](#), [fipa\\_envelope\\_HandleReceived\(\)](#), [index\\_compare\(\)](#), [index\\_find\(\)](#), [message\\_xml\\_parse\\_\\_message\(\)](#), [mxml\\_parse\\_element\(\)](#), [mxmlFindElement\(\)](#), [mxmlIndexNew\(\)](#), [scan\\_file\(\)](#), [sort\\_node\(\)](#), [type\\_cb\(\)](#), and [write\\_documentation\(\)](#).

#### 13.64.1.2 void [mxmlElementSetAttr](#) ([mxml\\_node\\_t](#) \* *node*, const char \* *name*, const char \* *value*)

Definition at line 90 of file mxml-attr.c.

References [mxml\\_value\\_s::attrs](#), [mxml\\_value\\_u::element](#), [MXML\\_ELEMENT](#), [mxml\\_error\(\)](#), [mxml\\_value\\_s::name](#), [mxml\\_attr\\_s::name](#), [mxml\\_value\\_s::num\\_attrs](#), [mxml\\_node\\_s::type](#), [mxml\\_attr\\_s::value](#), and [mxml\\_node\\_s::value](#).

Referenced by [add\\_variable\(\)](#), [agent\\_xml\\_compose\\_\\_agent\\_code\(\)](#), [agent\\_xml\\_compose\\_\\_create\\_row\\_nodes\(\)](#), [agent\\_xml\\_compose\\_\\_data\(\)](#), [agent\\_xml\\_compose\\_\\_message\(\)](#), [agent\\_xml\\_compose\\_\\_task\(\)](#), [agent\\_xml\\_compose\\_\\_tasks\(\)](#), [fipa\\_envelope\\_Compose\\_\\_params\(\)](#), [main\(\)](#), [mxml\\_parse\\_element\(\)](#), [scan\\_file\(\)](#), [sort\\_node\(\)](#), and [update\\_comment\(\)](#).



## 13.65 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mxml-2.2.2/mxml-entity.c File Reference

```
#include "config.h"
#include "mxml.h"
```

### Functions

- static [int default\\_callback](#) (const char \*name)
- [int mxmlEntityAddCallback](#) (int(\*cb)(const char \*name))
- const char \* [mxmlEntityGetName](#) (int val)
- [int mxmlEntityGetValue](#) (const char \*name)
- void [mxmlEntityRemoveCallback](#) (int(\*cb)(const char \*name))

### Variables

- static [int num\\_callbacks](#) = 1
- static [int\(\\* callbacks](#) [100])(const char \*name)

#### 13.65.1 Function Documentation

##### 13.65.1.1 static int default\_callback (const char \* *name*) [static]

Definition at line 169 of file mxml-entity.c.

References [int](#).

##### 13.65.1.2 int mxmlEntityAddCallback (int(\*) (const char \*name) *cb*)

Definition at line 65 of file mxml-entity.c.

References [callbacks](#), [mxml\\_error\(\)](#), and [num\\_callbacks](#).

##### 13.65.1.3 const char\* mxmlEntityGetName (int *val*)

Definition at line 91 of file mxml-entity.c.

Referenced by [mxml\\_write\\_name\(\)](#), and [mxml\\_write\\_string\(\)](#).

##### 13.65.1.4 int mxmlEntityGetValue (const char \* *name*)

Definition at line 121 of file mxml-entity.c.

References [callbacks](#), and [num\\_callbacks](#).

Referenced by [mxml\\_get\\_entity\(\)](#).

**13.65.1.5 void mxmlEntityRemoveCallback (int(\*) (const char \*name) cb)**

Definition at line 140 of file mxml-entity.c.

References `callbacks`, and `num_callbacks`.

**13.65.2 Variable Documentation****13.65.2.1 int(\* callbacks[100])(const char \*name) [static]**

**Initial value:**

```
{
    default_callback
}
```

Referenced by `mxmlEntityAddCallback()`, `mxmlEntityGetValue()`, and `mxmlEntityRemoveCallback()`.

**13.65.2.2 int num\_callbacks = 1 [static]**

Definition at line 53 of file mxml-entity.c.

Referenced by `mxmlEntityAddCallback()`, `mxmlEntityGetValue()`, and `mxmlEntityRemoveCallback()`.

## 13.66 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mxml-2.2.2/mxml-file.c File Reference

```
#include "config.h"
#include "mxml.h"
#include <unistd.h>
```

### Data Structures

- struct [mxml\\_fdbuf\\_s](#)

### Defines

- #define [ENCODE\\_UTF8](#) 0
- #define [ENCODE\\_UTF16BE](#) 1
- #define [ENCODE\\_UTF16LE](#) 2
- #define [mxml\\_bad\\_char](#)(ch) ((ch) < ' ' && (ch) != '\n' && (ch) != '\r' && (ch) != '\t')

### Typedefs

- typedef struct [mxml\\_fdbuf\\_s](#) [mxml\\_fdbuf\\_t](#)

### Functions

- static [int](#) [mxml\\_add\\_char](#) ([int](#) ch, [char](#) \*\*ptr, [char](#) \*\*buffer, [int](#) \*bufsize)
- static [int](#) [mxml\\_fd\\_getc](#) ([void](#) \*p, [int](#) \*encoding)
- static [int](#) [mxml\\_fd\\_putc](#) ([int](#) ch, [void](#) \*p)
- static [int](#) [mxml\\_fd\\_read](#) ([mxml\\_fdbuf\\_t](#) \*buf)
- static [int](#) [mxml\\_fd\\_write](#) ([mxml\\_fdbuf\\_t](#) \*buf)
- static [int](#) [mxml\\_file\\_getc](#) ([void](#) \*p, [int](#) \*encoding)
- static [int](#) [mxml\\_file\\_putc](#) ([int](#) ch, [void](#) \*p)
- static [int](#) [mxml\\_get\\_entity](#) ([mxml\\_node\\_t](#) \*parent, [void](#) \*p, [int](#) \*encoding, [int](#)(\*getc\_cb)([void](#) \*, [int](#) \*))
- static [mxml\\_node\\_t](#) \* [mxml\\_load\\_data](#) ([mxml\\_node\\_t](#) \*top, [void](#) \*p, [mxml\\_type\\_t](#)(\*cb)([mxml\\_node\\_t](#) \*), [int](#)(\*getc\_cb)([void](#) \*, [int](#) \*))
- static [int](#) [mxml\\_parse\\_element](#) ([mxml\\_node\\_t](#) \*node, [void](#) \*p, [int](#) \*encoding, [int](#)(\*getc\_cb)([void](#) \*, [int](#) \*))
- static [int](#) [mxml\\_string\\_getc](#) ([void](#) \*p, [int](#) \*encoding)
- static [int](#) [mxml\\_string\\_putc](#) ([int](#) ch, [void](#) \*p)
- static [int](#) [mxml\\_write\\_name](#) ([const](#) [char](#) \*s, [void](#) \*p, [int](#)(\*putc\_cb)([int](#), [void](#) \*))
- static [int](#) [mxml\\_write\\_node](#) ([mxml\\_node\\_t](#) \*node, [void](#) \*p, [const](#) [char](#) \*(\*cb)([mxml\\_node\\_t](#) \*, [int](#)), [int](#) col, [int](#)(\*putc\_cb)([int](#), [void](#) \*))
- static [int](#) [mxml\\_write\\_string](#) ([const](#) [char](#) \*s, [void](#) \*p, [int](#)(\*putc\_cb)([int](#), [void](#) \*))
- static [int](#) [mxml\\_write\\_ws](#) ([mxml\\_node\\_t](#) \*node, [void](#) \*p, [const](#) [char](#) \*(\*cb)([mxml\\_node\\_t](#) \*, [int](#)), [int](#) ws, [int](#) col, [int](#)(\*putc\_cb)([int](#), [void](#) \*))
- [mxml\\_node\\_t](#) \* [mxmlLoadFd](#) ([mxml\\_node\\_t](#) \*top, [int](#) fd, [mxml\\_type\\_t](#)(\*cb)([mxml\\_node\\_t](#) \*node))
- [mxml\\_node\\_t](#) \* [mxmlLoadFile](#) ([mxml\\_node\\_t](#) \*top, [FILE](#) \*fp, [mxml\\_type\\_t](#)(\*cb)([mxml\\_node\\_t](#) \*node))

- `mxml_node_t * mxmlloadString (mxml_node_t *top, const char *s, mxml_type_t(*cb)(mxml_node_t *node))`
- `char * mxmlSaveAllocString (mxml_node_t *node, const char *(*cb)(mxml_node_t *node, int ws))`
- `int mxmlSaveFd (mxml_node_t *node, int fd, const char *(*cb)(mxml_node_t *node, int ws))`
- `int mxmlSaveFile (mxml_node_t *node, FILE *fp, const char *(*cb)(mxml_node_t *node, int ws))`
- `int mxmlSaveString (mxml_node_t *node, char *buffer, int bufsize, const char *(*cb)(mxml_node_t *node, int ws))`
- `void mxmlSetCustomHandlers (mxml_custom_load_cb_t load, mxml_custom_save_cb_t save)`
- `void mxmlSetErrorCallback (void(*cb)(const char *))`

## Variables

- `void(* mxml_error_cb )(const char *)`
- `static mxml_custom_load_cb_t mxml_custom_load_cb = NULL`
- `static mxml_custom_save_cb_t mxml_custom_save_cb = NULL`

### 13.66.1 Define Documentation

#### 13.66.1.1 `#define ENCODE_UTF16BE 1`

Definition at line 68 of file `mxml-file.c`.

Referenced by `mxml_fd_getc()`, `mxml_file_getc()`, and `mxml_string_getc()`.

#### 13.66.1.2 `#define ENCODE_UTF16LE 2`

Definition at line 69 of file `mxml-file.c`.

Referenced by `mxml_fd_getc()`, `mxml_file_getc()`, and `mxml_string_getc()`.

#### 13.66.1.3 `#define ENCODE_UTF8 0`

Definition at line 67 of file `mxml-file.c`.

Referenced by `mxml_fd_getc()`, `mxml_file_getc()`, `mxml_load_data()`, and `mxml_string_getc()`.

#### 13.66.1.4 `#define mxml_bad_char(ch) ((ch) < ' ' && (ch) != '\n' && (ch) != '\r' && (ch) != '\t')`

Definition at line 76 of file `mxml-file.c`.

Referenced by `mxml_fd_getc()`, `mxml_file_getc()`, `mxml_get_entity()`, and `mxml_string_getc()`.

### 13.66.2 Typedef Documentation

#### 13.66.2.1 `typedef struct mxml_fdbuf_s mxml_fdbuf_t`

### 13.66.3 Function Documentation

#### 13.66.3.1 `static int mxml_add_char (int ch, char **ptr, char **buffer, int *bufsize) [static]`

Definition at line 483 of file `mxml-file.c`.

References `mxml_error()`.

Referenced by `mxml_load_data()`, and `mxml_parse_element()`.

### 13.66.3.2 `static int mxml_fd_getc (void *p, int *encoding) [static]`

Definition at line 563 of file `mxml-file.c`.

References `buf`, `mxml_fdbuf_s::current`, `ENCODE_UTF16BE`, `ENCODE_UTF16LE`, `ENCODE_UTF8`, `mxml_fdbuf_s::end`, `mxml_bad_char`, `mxml_error()`, and `mxml_fd_read()`.

Referenced by `mxmlLoadFd()`.

### 13.66.3.3 `static int mxml_fd_putc (int ch, void *p) [static]`

Definition at line 850 of file `mxml-file.c`.

References `buf`, `mxml_fdbuf_s::current`, `mxml_fdbuf_s::end`, and `mxml_fd_write()`.

Referenced by `mxmlSaveFd()`.

### 13.66.3.4 `static int mxml_fd_read (mxml_fdbuf_t *buf) [static]`

Definition at line 919 of file `mxml-file.c`.

References `mxml_fdbuf_s::buffer`, `mxml_fdbuf_s::current`, `mxml_fdbuf_s::end`, and `mxml_fdbuf_s::fd`.

Referenced by `mxml_fd_getc()`.

### 13.66.3.5 `static int mxml_fd_write (mxml_fdbuf_t *buf) [static]`

Definition at line 958 of file `mxml-file.c`.

References `mxml_fdbuf_s::buffer`, `mxml_fdbuf_s::current`, and `mxml_fdbuf_s::fd`.

Referenced by `mxml_fd_putc()`, and `mxmlSaveFd()`.

### 13.66.3.6 `static int mxml_file_getc (void *p, int *encoding) [static]`

Definition at line 1001 of file `mxml-file.c`.

References `ENCODE_UTF16BE`, `ENCODE_UTF16LE`, `ENCODE_UTF8`, `mxml_bad_char`, and `mxml_error()`.

Referenced by `mxmlLoadFile()`.

### 13.66.3.7 `static int mxml_file_putc (int ch, void *p) [static]`

Definition at line 1200 of file `mxml-file.c`.

Referenced by `mxmlSaveFile()`.

**13.66.3.8** `static int mxml_get_entity (mxml_node_t * parent, void * p, int * encoding, int(*)(void *, int *) getc_cb) [static]`

Definition at line 1255 of file mxml-file.c.

References `mxml_value_u::element`, `mxml_bad_char`, `mxml_error()`, `mxmlEntityGetValue()`, `mxml_value_s::name`, and `mxml_node_s::value`.

Referenced by `mxml_load_data()`, and `mxml_parse_element()`.

**13.66.3.9** `static mxml_node_t * mxml_load_data (mxml_node_t * top, void * p, mxml_type_t(*)(mxml_node_t *) cb, int(*)(void *, int *) getc_cb) [static]`

Definition at line 1316 of file mxml-file.c.

References `mxml_value_u::element`, `ENCODE_UTF8`, `mxml_add_char()`, `MXML_CUSTOM`, `mxml_custom_load_cb`, `mxml_error()`, `mxml_get_entity()`, `MXML_INTEGER`, `MXML_OPAQUE`, `mxml_parse_element()`, `MXML_REAL`, `MXML_TEXT`, `mxmlDelete()`, `mxmlNewCustom()`, `mxmlNewElement()`, `mxmlNewInteger()`, `mxmlNewOpaque()`, `mxmlNewReal()`, `mxmlNewText()`, `mxml_value_s::name`, `mxml_node_s::parent`, and `mxml_node_s::value`.

Referenced by `mxmlLoadFd()`, `mxmlLoadFile()`, and `mxmlLoadString()`.

**13.66.3.10** `static int mxml_parse_element (mxml_node_t * node, void * p, int * encoding, int(*)(void *, int *) getc_cb) [static]`

Definition at line 1840 of file mxml-file.c.

References `mxml_value_u::element`, `mxml_add_char()`, `mxml_error()`, `mxml_get_entity()`, `mxmlElementGetAttr()`, `mxmlElementSetAttr()`, `mxml_value_s::name`, and `mxml_node_s::value`.

Referenced by `mxml_load_data()`.

**13.66.3.11** `static int mxml_string_getc (void * p, int * encoding) [static]`

Definition at line 2102 of file mxml-file.c.

References `ENCODE_UTF16BE`, `ENCODE_UTF16LE`, `ENCODE_UTF8`, `mxml_bad_char`, and `mxml_error()`.

Referenced by `mxmlLoadString()`.

**13.66.3.12** `static int mxml_string_putc (int ch, void * p) [static]`

Definition at line 2338 of file mxml-file.c.

Referenced by `mxmlSaveString()`.

**13.66.3.13** `static int mxml_write_name (const char * s, void * p, int(*)(int, void *) putc_cb) [static]`

Definition at line 2412 of file mxml-file.c.

References `mxmlEntityGetName()`.

Referenced by `mxml_write_node()`.

**13.66.3.14** `static int mxml_write_node (mxml_node_t * node, void * p, const char  
 *(*)(mxml_node_t *, int) cb, int col, int(*)(int, void *) putc_cb) [static]`

Definition at line 2487 of file mxml-file.c.

References `mxml_value_s::attrs`, `mxml_node_s::child`, `mxml_value_u::element`, `mxml_value_u::integer`, `MXML_CUSTOM`, `mxml_custom_save_cb`, `MXML_ELEMENT`, `MXML_INTEGER`, `MXML_OPAQUE`, `MXML_REAL`, `MXML_TEXT`, `MXML_WRAP`, `mxml_write_name()`, `mxml_write_string()`, `mxml_write_ws()`, `MXML_WS_AFTER_CLOSE`, `MXML_WS_AFTER_OPEN`, `MXML_WS_BEFORE_CLOSE`, `MXML_WS_BEFORE_OPEN`, `mxml_attr_s::name`, `mxml_value_s::name`, `mxml_node_s::next`, `mxml_value_s::num_attrs`, `mxml_value_u::opaque`, `mxml_node_s::prev`, `mxml_value_u::real`, `mxml_text_s::string`, `mxml_value_u::text`, `mxml_node_s::type`, `mxml_attr_s::value`, `mxml_node_s::value`, and `mxml_text_s::whitespace`.

Referenced by `mxmlSaveFd()`, `mxmlSaveFile()`, and `mxmlSaveString()`.

**13.66.3.15** `static int mxml_write_string (const char * s, void * p, int(*)(int, void *) putc_cb)  
 [static]`

Definition at line 2768 of file mxml-file.c.

References `mxmlEntityGetName()`.

Referenced by `mxml_write_node()`.

**13.66.3.16** `static int mxml_write_ws (mxml_node_t * node, void * p, const char *(*)(mxml_node_t  
 *, int) cb, int ws, int col, int(*)(int, void *) putc_cb) [static]`

Definition at line 2808 of file mxml-file.c.

References `MXML_TAB`.

Referenced by `mxml_write_node()`.

**13.66.3.17** `mxml_node_t* mxmlLoadFd (mxml_node_t * top, int fd,  
 mxml_type_t(*)(mxml_node_t *node) cb)`

Definition at line 159 of file mxml-file.c.

References `buf`, `mxml_fdbuf_s::buffer`, `mxml_fdbuf_s::current`, `mxml_fdbuf_s::end`, `mxml_fdbuf_s::fd`, `mxml_fd_getc()`, and `mxml_load_data()`.

Referenced by `main()`.

**13.66.3.18** `mxml_node_t* mxmlLoadFile (mxml_node_t * top, FILE * fp,  
 mxml_type_t(*)(mxml_node_t *node) cb)`

Definition at line 199 of file mxml-file.c.

References `mxml_file_getc()`, and `mxml_load_data()`.

Referenced by `main()`.

### 13.66.3.19 **mxmxml\_node\_t\* mxmlLoadString (mxmxml\_node\_t \* *top*, const char \* *s*, mxml\_type\_t (\*)(mxmxml\_node\_t \**node*) *cb*)**

Definition at line 228 of file mxml-file.c.

References mxml\_load\_data(), and mxml\_string\_getc().

Referenced by acc\_connection\_Thread(), agent\_xml\_compose(), fipa\_envelope\_Compose(), fipa\_envelope\_Parse(), main(), MC\_LoadAgentFromFile(), message\_InitializeFromConnection(), and message\_InitializeFromString().

### 13.66.3.20 **char\* mxmlSaveAllocString (mxmxml\_node\_t \* *node*, const char \*\* (mxmxml\_node\_t \**node*, int *ws*) *cb*)**

Definition at line 258 of file mxml-file.c.

References mxmlSaveString().

Referenced by fipa\_envelope\_Compose(), MC\_GetAgentXMLString(), and message\_InitializeFromAgent().

### 13.66.3.21 **int mxmlSaveFd (mxmxml\_node\_t \* *node*, int *fd*, const char \*\* (mxmxml\_node\_t \**node*, int *ws*) *cb*)**

Definition at line 315 of file mxml-file.c.

References buf, mxml\_fdbuf\_s::buffer, mxml\_fdbuf\_s::current, mxml\_fdbuf\_s::end, mxml\_fdbuf\_s::fd, mxml\_fd\_putc(), mxml\_fd\_write(), and mxml\_write\_node().

Referenced by main().

### 13.66.3.22 **int mxmlSaveFile (mxmxml\_node\_t \* *node*, FILE \* *fp*, const char \*\* (mxmxml\_node\_t \**node*, int *ws*) *cb*)**

Definition at line 362 of file mxml-file.c.

References mxml\_file\_putc(), and mxml\_write\_node().

Referenced by main().

### 13.66.3.23 **int mxmlSaveString (mxmxml\_node\_t \* *node*, char \* *buffer*, int *bufsize*, const char \*\* (mxmxml\_node\_t \**node*, int *ws*) *cb*)**

Definition at line 404 of file mxml-file.c.

References mxml\_string\_putc(), and mxml\_write\_node().

Referenced by main(), and mxmlSaveAllocString().

### 13.66.3.24 **void mxmlSetCustomHandlers (mxml\_custom\_load\_cb\_t *load*, mxml\_custom\_save\_cb\_t *save*)**

Definition at line 456 of file mxml-file.c.

References mxml\_custom\_load\_cb, and mxml\_custom\_save\_cb.



### 13.66.3.25 void mxmlSetErrorCallback (void(\*) (const char \*) cb)

Definition at line 471 of file mxml-file.c.

References mxml\_error\_cb.

## 13.66.4 Variable Documentation

### 13.66.4.1 mxml\_custom\_load\_cb\_t mxml\_custom\_load\_cb = NULL [static]

Definition at line 103 of file mxml-file.c.

Referenced by mxml\_load\_data(), and mxmlSetCustomHandlers().

### 13.66.4.2 mxml\_custom\_save\_cb\_t mxml\_custom\_save\_cb = NULL [static]

Definition at line 104 of file mxml-file.c.

Referenced by mxml\_write\_node(), and mxmlSetCustomHandlers().

### 13.66.4.3 void(\*) mxml\_error\_cb (const char \*)

Referenced by mxml\_error(), and mxmlSetErrorCallback().

## 13.67 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mxml-2.2.2/mxml-index.c File Reference

```
#include "config.h"
#include "mxml.h"
```

### Functions

- static [int](#) [index\\_compare](#) ([mxml\\_index\\_t](#) \*ind, [mxml\\_node\\_t](#) \*first, [mxml\\_node\\_t](#) \*second)
- static [int](#) [index\\_find](#) ([mxml\\_index\\_t](#) \*ind, const char \*element, const char \*value, [mxml\\_node\\_t](#) \*node)
- static void [index\\_sort](#) ([mxml\\_index\\_t](#) \*ind, [int](#) left, [int](#) right)
- void [mxmlIndexDelete](#) ([mxml\\_index\\_t](#) \*ind)
- [mxml\\_node\\_t](#) \* [mxmlIndexEnum](#) ([mxml\\_index\\_t](#) \*ind)
- [mxml\\_node\\_t](#) \* [mxmlIndexFind](#) ([mxml\\_index\\_t](#) \*ind, const char \*element, const char \*value)
- [mxml\\_index\\_t](#) \* [mxmlIndexNew](#) ([mxml\\_node\\_t](#) \*node, const char \*element, const char \*attr)
- [mxml\\_node\\_t](#) \* [mxmlIndexReset](#) ([mxml\\_index\\_t](#) \*ind)

### 13.67.1 Function Documentation

**13.67.1.1** static [int](#) [index\\_compare](#) ([mxml\\_index\\_t](#) \* *ind*, [mxml\\_node\\_t](#) \* *first*, [mxml\\_node\\_t](#) \* *second*) [**static**]

Definition at line 494 of file [mxml-index.c](#).

References [mxml\\_index\\_s::attr](#), [mxml\\_value\\_u::element](#), [mxmlElementGetAttr\(\)](#), [mxml\\_value\\_s::name](#), and [mxml\\_node\\_s::value](#).

Referenced by [index\\_sort\(\)](#).

**13.67.1.2** static [int](#) [index\\_find](#) ([mxml\\_index\\_t](#) \* *ind*, const char \* *element*, const char \* *value*, [mxml\\_node\\_t](#) \* *node*) [**static**]

Definition at line 533 of file [mxml-index.c](#).

References [mxml\\_index\\_s::attr](#), [mxml\\_value\\_u::element](#), [mxmlElementGetAttr\(\)](#), [mxml\\_value\\_s::name](#), and [mxml\\_node\\_s::value](#).

Referenced by [mxmlIndexFind\(\)](#).

**13.67.1.3** static void [index\\_sort](#) ([mxml\\_index\\_t](#) \* *ind*, [int](#) *left*, [int](#) *right*) [**static**]

Definition at line 576 of file [mxml-index.c](#).

References [index\\_compare\(\)](#), and [mxml\\_index\\_s::nodes](#).

Referenced by [mxmlIndexNew\(\)](#).

**13.67.1.4** void [mxmlIndexDelete](#) ([mxml\\_index\\_t](#) \* *ind*)

Definition at line 58 of file [mxml-index.c](#).

References mxml\_index\_s::alloc\_nodes, mxml\_index\_s::attr, and mxml\_index\_s::nodes.

Referenced by main(), and mxmlIndexNew().

#### 13.67.1.5 mxml\_node\_t\* mxmlIndexEnum (mxml\_index\_t \* *ind*)

Definition at line 88 of file mxml-index.c.

References mxml\_index\_s::cur\_node, mxml\_index\_s::nodes, and mxml\_index\_s::num\_nodes.

Referenced by main(), and mxmlIndexFind().

#### 13.67.1.6 mxml\_node\_t\* mxmlIndexFind (mxml\_index\_t \* *ind*, const char \* *element*, const char \* *value*)

Definition at line 118 of file mxml-index.c.

References mxml\_index\_s::attr, mxml\_index\_s::cur\_node, index\_find(), mxmlIndexEnum(), mxml\_index\_s::nodes, and mxml\_index\_s::num\_nodes.

Referenced by main().

#### 13.67.1.7 mxml\_index\_t\* mxmlIndexNew (mxml\_node\_t \* *node*, const char \* *element*, const char \* *attr*)

Definition at line 301 of file mxml-index.c.

References mxml\_index\_s::alloc\_nodes, mxml\_index\_s::attr, mxml\_value\_u::element, index\_sort(), MXML\_DESCEND, mxml\_error(), mxmlElementGetAttr(), mxmlFindElement(), mxmlIndexDelete(), mxml\_value\_s::name, mxml\_index\_s::nodes, mxml\_index\_s::num\_nodes, and mxml\_node\_s::value.

Referenced by main().

#### 13.67.1.8 mxml\_node\_t\* mxmlIndexReset (mxml\_index\_t \* *ind*)

Definition at line 459 of file mxml-index.c.

References mxml\_index\_s::cur\_node, mxml\_index\_s::nodes, and mxml\_index\_s::num\_nodes.

Referenced by main().

## 13.68 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mxml-2.2.2/mxml-node.c File Reference

```
#include "config.h"
#include "mxml.h"
```

### Functions

- static `mxml_node_t * mxml_new (mxml_node_t *parent, mxml_type_t type)`
- void `mxmlAdd (mxml_node_t *parent, int where, mxml_node_t *child, mxml_node_t *node)`
- void `mxmlDelete (mxml_node_t *node)`
- `mxml_node_t * mxmlNewCustom (mxml_node_t *parent, void *data, void(*destroy)(void *))`
- `mxml_node_t * mxmlNewElement (mxml_node_t *parent, const char *name)`
- `mxml_node_t * mxmlNewInteger (mxml_node_t *parent, int integer)`
- `mxml_node_t * mxmlNewOpaque (mxml_node_t *parent, const char *opaque)`
- `mxml_node_t * mxmlNewReal (mxml_node_t *parent, double real)`
- `mxml_node_t * mxmlNewText (mxml_node_t *parent, int whitespace, const char *string)`
- `mxml_node_t * mxmlNewTextf (mxml_node_t *parent, int whitespace, const char *format,...)`
- void `mxmlRemove (mxml_node_t *node)`

### 13.68.1 Function Documentation

#### 13.68.1.1 static `mxml_node_t * mxml_new (mxml_node_t * parent, mxml_type_t type)` [static]

Definition at line 617 of file mxml-node.c.

References MXML\_ADD\_AFTER, MXML\_ADD\_TO\_PARENT, mxmlAdd(), node, and mxml\_node\_s::type.

Referenced by mxmlNewCustom(), mxmlNewElement(), mxmlNewInteger(), mxmlNewOpaque(), mxmlNewReal(), mxmlNewText(), and mxmlNewTextf().

#### 13.68.1.2 void `mxmlAdd (mxml_node_t * parent, int where, mxml_node_t * child, mxml_node_t * node)`

Definition at line 62 of file mxml-node.c.

References mxml\_node\_s::child, mxml\_node\_s::last\_child, MXML\_ADD\_AFTER, MXML\_ADD\_BEFORE, mxmlRemove(), mxml\_node\_s::next, mxml\_node\_s::parent, and mxml\_node\_s::prev.

Referenced by add\_variable(), agent\_xml\_compose(), agent\_xml\_compose\_\_agent\_data(), agent\_xml\_compose\_\_create\_row\_nodes(), agent\_xml\_compose\_\_data(), agent\_xml\_compose\_\_gaf\_message(), agent\_xml\_compose\_\_message(), agent\_xml\_compose\_\_mobile\_agent(), agent\_xml\_compose\_\_task(), agent\_xml\_compose\_\_tasks(), fipa\_envelope\_Compose(), fipa\_envelope\_Compose\_\_envelope(), fipa\_envelope\_Compose\_\_params(), mxml\_new(), scan\_file(), and sort\_node().

#### 13.68.1.3 void `mxmlDelete (mxml_node_t * node)`

Definition at line 196 of file mxml-node.c.

References mxml\_value\_s::attrs, mxml\_node\_s::child, mxml\_value\_u::custom, mxml\_custom\_s::data, mxml\_custom\_s::destroy, mxml\_value\_u::element, MXML\_CUSTOM, MXML\_ELEMENT, MXML\_INTEGER, MXML\_OPAQUE, MXML\_REAL, MXML\_TEXT, mxmlDelete(), mxmlRemove(), mxml\_attr\_s::name, mxml\_value\_s::name, mxml\_value\_s::num\_attrs, mxml\_value\_u::opaque, mxml\_text\_s::string, mxml\_value\_u::text, mxml\_node\_s::type, mxml\_attr\_s::value, and mxml\_node\_s::value.

Referenced by add\_variable(), agent\_datastate\_Destroy(), fipa\_envelope\_Compose(), fipa\_envelope\_Parse(), main(), message\_Destroy(), mxml\_load\_data(), mxmlDelete(), scan\_file(), and sort\_node().

#### 13.68.1.4 mxml\_node\_t\* mxmlNewCustom (mxml\_node\_t \* *parent*, void \* *data*, void(\*) (void \*) *destroy*)

Definition at line 287 of file mxml-node.c.

References mxml\_value\_u::custom, mxml\_custom\_s::data, mxml\_custom\_s::destroy, MXML\_CUSTOM, mxml\_new(), node, and mxml\_node\_s::value.

Referenced by mxml\_load\_data().

#### 13.68.1.5 mxml\_node\_t\* mxmlNewElement (mxml\_node\_t \* *parent*, const char \* *name*)

Definition at line 323 of file mxml-node.c.

References mxml\_value\_u::element, MXML\_ELEMENT, mxml\_new(), mxml\_value\_s::name, node, and mxml\_node\_s::value.

Referenced by add\_variable(), agent\_xml\_compose\_\_agent\_code(), agent\_xml\_compose\_\_agent\_data(), agent\_xml\_compose\_\_create\_row\_nodes(), agent\_xml\_compose\_\_data(), agent\_xml\_compose\_\_gaf\_message(), agent\_xml\_compose\_\_home(), agent\_xml\_compose\_\_message(), agent\_xml\_compose\_\_mobile\_agent(), agent\_xml\_compose\_\_name(), agent\_xml\_compose\_\_owner(), agent\_xml\_compose\_\_task(), agent\_xml\_compose\_\_tasks(), agent\_xml\_compose\_\_wg\_code(), fipa\_envelope\_Compose\_\_acl\_representation(), fipa\_envelope\_Compose\_\_date(), fipa\_envelope\_Compose\_\_envelope(), fipa\_envelope\_Compose\_\_from(), fipa\_envelope\_Compose\_\_intended\_receiver(), fipa\_envelope\_Compose\_\_params(), fipa\_envelope\_Compose\_\_payload\_encoding(), fipa\_envelope\_Compose\_\_to(), main(), mxml\_load\_data(), scan\_file(), and xml\_new\_cdata().

#### 13.68.1.6 mxml\_node\_t\* mxmlNewInteger (mxml\_node\_t \* *parent*, int *integer*)

Definition at line 361 of file mxml-node.c.

References mxml\_value\_u::integer, MXML\_INTEGER, mxml\_new(), node, and mxml\_node\_s::value.

Referenced by main(), and mxml\_load\_data().

#### 13.68.1.7 mxml\_node\_t\* mxmlNewOpaque (mxml\_node\_t \* *parent*, const char \* *opaque*)

Definition at line 392 of file mxml-node.c.

References mxml\_new(), MXML\_OPAQUE, node, mxml\_value\_u::opaque, and mxml\_node\_s::value.

Referenced by main(), and mxml\_load\_data().

#### 13.68.1.8 mxml\_node\_t\* mxmlNewReal (mxml\_node\_t \* *parent*, double *real*)

Definition at line 430 of file mxml-node.c.

References `mxml_new()`, `MXML_REAL`, `node`, `mxml_value_u::real`, and `mxml_node_s::value`.

Referenced by `main()`, and `mxml_load_data()`.

#### **13.68.1.9 `mxml_node_t*` `mxmlNewText` (`mxml_node_t` \* *parent*, `int` *whitespace*, `const char` \* *string*)**

Definition at line 462 of file `mxml-node.c`.

References `mxml_new()`, `MXML_TEXT`, `node`, `mxml_text_s::string`, `mxml_value_u::text`, `mxml_node_s::value`, and `mxml_text_s::whitespace`.

Referenced by `agent_xml_compose__create_row_nodes()`, `agent_xml_compose__home()`, `agent_xml_compose__name()`, `agent_xml_compose__owner()`, `agent_xml_compose__wg_code()`, `fipa_envelope_Compose__acl_representation()`, `fipa_envelope_Compose__date()`, `fipa_envelope_Compose__from()`, `fipa_envelope_Compose__intended_receiver()`, `fipa_envelope_Compose__payload_encoding()`, `fipa_envelope_Compose__to()`, `main()`, `mxml_load_data()`, and `scan_file()`.

#### **13.68.1.10 `mxml_node_t*` `mxmlNewTextf` (`mxml_node_t` \* *parent*, `int` *whitespace*, `const char` \* *format*, ...)**

Definition at line 506 of file `mxml-node.c`.

References `mxml_new()`, `mxml_strdupf()`, `MXML_TEXT`, `node`, `mxml_text_s::string`, `mxml_value_u::text`, `mxml_node_s::value`, and `mxml_text_s::whitespace`.

#### **13.68.1.11 `void` `mxmlRemove` (`mxml_node_t` \* *node*)**

Definition at line 553 of file `mxml-node.c`.

References `mxml_node_s::child`, `mxml_node_s::last_child`, `mxml_node_s::next`, `mxml_node_s::parent`, and `mxml_node_s::prev`.

Referenced by `mxmlAdd()`, and `mxmlDelete()`.

## 13.69 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mxml-2.2.2/mxml-private.c File Reference

```
#include "config.h"
#include "mxml.h"
```

### Functions

- void [mxml\\_error](#) (const char \*format,...)
- [mxml\\_type\\_t mxml\\_integer\\_cb](#) ([mxml\\_node\\_t](#) \*node)
- [mxml\\_type\\_t mxml\\_opaque\\_cb](#) ([mxml\\_node\\_t](#) \*node)
- [mxml\\_type\\_t mxml\\_real\\_cb](#) ([mxml\\_node\\_t](#) \*node)

### Variables

- void(\* [mxml\\_error\\_cb](#) )(const char \*) = NULL

### 13.69.1 Function Documentation

#### 13.69.1.1 void [mxml\\_error](#) (const char \**format*, ...)

Definition at line 49 of file mxml-private.c.

References [mxml\\_error\\_cb](#), and [mxml\\_strdup\(\)](#).

Referenced by [mxml\\_add\\_char\(\)](#), [mxml\\_fd\\_getc\(\)](#), [mxml\\_file\\_getc\(\)](#), [mxml\\_get\\_entity\(\)](#), [mxml\\_load\\_data\(\)](#), [mxml\\_parse\\_element\(\)](#), [mxml\\_string\\_getc\(\)](#), [mxmlElementSetAttr\(\)](#), [mxmlEntityAddCallback\(\)](#), and [mxmlIndexNew\(\)](#).

#### 13.69.1.2 [mxml\\_type\\_t mxml\\_integer\\_cb](#) ([mxml\\_node\\_t](#) \**node*)

Definition at line 95 of file mxml-private.c.

References [MXML\\_INTEGER](#).

#### 13.69.1.3 [mxml\\_type\\_t mxml\\_opaque\\_cb](#) ([mxml\\_node\\_t](#) \**node*)

Definition at line 108 of file mxml-private.c.

References [MXML\\_OPAQUE](#).

#### 13.69.1.4 [mxml\\_type\\_t mxml\\_real\\_cb](#) ([mxml\\_node\\_t](#) \**node*)

Definition at line 121 of file mxml-private.c.

References [MXML\\_REAL](#).

## 13.69.2 Variable Documentation

### 13.69.2.1 `void(* mxml_error_cb)(const char *) = NULL`

Referenced by `mxml_error()`, and `mxmlSetErrorCallback()`.



## 13.70 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mxml-2.2.2/mxml-search.c File Reference

```
#include "config.h"
#include "mxml.h"
```

### Functions

- [mxml\\_node\\_t \\* mxmlFindElement](#) ([mxml\\_node\\_t \\* node](#), [mxml\\_node\\_t \\* top](#), [const char \\* name](#), [const char \\* attr](#), [const char \\* value](#), [int descend](#))
- [mxml\\_node\\_t \\* mxmlWalkNext](#) ([mxml\\_node\\_t \\* node](#), [mxml\\_node\\_t \\* top](#), [int descend](#))
- [mxml\\_node\\_t \\* mxmlWalkPrev](#) ([mxml\\_node\\_t \\* node](#), [mxml\\_node\\_t \\* top](#), [int descend](#))

### 13.70.1 Function Documentation

#### 13.70.1.1 [mxml\\_node\\_t \\* mxmlFindElement](#) ([mxml\\_node\\_t \\* node](#), [mxml\\_node\\_t \\* top](#), [const char \\* name](#), [const char \\* attr](#), [const char \\* value](#), [int descend](#))

Definition at line 51 of file mxml-search.c.

References [mxml\\_value\\_u::element](#), [MXML\\_DESCEND](#), [MXML\\_ELEMENT](#), [mxmlElementGetAttr\(\)](#), [mxmlWalkNext\(\)](#), [mxml\\_value\\_s::name](#), [mxml\\_node\\_s::next](#), [mxml\\_node\\_s::type](#), and [mxml\\_node\\_s::value](#).

Referenced by [agent\\_xml\\_parse\\_\\_fill\\_row\\_data\(\)](#), [agent\\_xml\\_parse\\_\\_task\(\)](#), [agent\\_xml\\_parse\\_\\_tasks\(\)](#), [fipa\\_envelope\\_HandleAclRepresentation\(\)](#), [fipa\\_envelope\\_HandleComments\(\)](#), [fipa\\_envelope\\_HandleDate\(\)](#), [fipa\\_envelope\\_HandleEnvelope\(\)](#), [fipa\\_envelope\\_HandleFrom\(\)](#), [fipa\\_envelope\\_HandleIntendedReceiver\(\)](#), [fipa\\_envelope\\_HandleParams\(\)](#), [fipa\\_envelope\\_HandlePayloadEncoding\(\)](#), [fipa\\_envelope\\_HandlePayloadLength\(\)](#), [fipa\\_envelope\\_HandleReceived\(\)](#), [fipa\\_envelope\\_HandleTo\(\)](#), [fipa\\_envelope\\_ParseAddresses\(\)](#), [fipa\\_envelope\\_ParseAgentIdentifier\(\)](#), [fipa\\_envelope\\_ParseResolvers\(\)](#), [main\(\)](#), [MC\\_LoadAgentFromFile\(\)](#), [message\\_xml\\_parse\(\)](#), [mxmlIndexNew\(\)](#), [scan\\_file\(\)](#), [sort\\_node\(\)](#), [write\\_documentation\(\)](#), [write\\_element\(\)](#), [xml\\_find\\_sibling\(\)](#), and [xml\\_get\\_child\(\)](#).

#### 13.70.1.2 [mxml\\_node\\_t \\* mxmlWalkNext](#) ([mxml\\_node\\_t \\* node](#), [mxml\\_node\\_t \\* top](#), [int descend](#))

Definition at line 133 of file mxml-search.c.

References [mxml\\_node\\_s::child](#), [mxml\\_node\\_s::next](#), and [mxml\\_node\\_s::parent](#).

Referenced by [mxmlFindElement\(\)](#), and [write\\_element\(\)](#).

#### 13.70.1.3 [mxml\\_node\\_t \\* mxmlWalkPrev](#) ([mxml\\_node\\_t \\* node](#), [mxml\\_node\\_t \\* top](#), [int descend](#))

Definition at line 169 of file mxml-search.c.

References [mxml\\_node\\_s::last\\_child](#), [mxml\\_node\\_s::parent](#), and [mxml\\_node\\_s::prev](#).

## 13.71 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mxml-2.2.2/mxml-set.c File Reference

```
#include "config.h"
#include "mxml.h"
```

### Functions

- [int mxmlSetCustom](#) ([mxml\\_node\\_t](#) \**node*, void \**data*, void(\**destroy*)(void \*))
- [int mxmlSetElement](#) ([mxml\\_node\\_t](#) \**node*, const char \**name*)
- [int mxmlSetInteger](#) ([mxml\\_node\\_t](#) \**node*, [int](#) *integer*)
- [int mxmlSetOpaque](#) ([mxml\\_node\\_t](#) \**node*, const char \**opaque*)
- [int mxmlSetReal](#) ([mxml\\_node\\_t](#) \**node*, double *real*)
- [int mxmlSetText](#) ([mxml\\_node\\_t](#) \**node*, [int](#) *whitespace*, const char \**string*)
- [int mxmlSetTextf](#) ([mxml\\_node\\_t](#) \**node*, [int](#) *whitespace*, const char \**format*,...)

### 13.71.1 Function Documentation

#### 13.71.1.1 [int mxmlSetCustom](#) ([mxml\\_node\\_t](#) \* *node*, void \* *data*, void(\*) (void \*) *destroy*)

Definition at line 46 of file mxml-set.c.

References [mxml\\_value\\_u::custom](#), [mxml\\_custom\\_s::data](#), [mxml\\_custom\\_s::destroy](#), [MXML\\_CUSTOM](#), [mxml\\_node\\_s::type](#), and [mxml\\_node\\_s::value](#).

#### 13.71.1.2 [int mxmlSetElement](#) ([mxml\\_node\\_t](#) \* *node*, const char \* *name*)

Definition at line 79 of file mxml-set.c.

References [mxml\\_value\\_u::element](#), [MXML\\_ELEMENT](#), [mxml\\_value\\_s::name](#), [mxml\\_node\\_s::type](#), and [mxml\\_node\\_s::value](#).

#### 13.71.1.3 [int mxmlSetInteger](#) ([mxml\\_node\\_t](#) \* *node*, [int](#) *integer*)

Definition at line 109 of file mxml-set.c.

References [mxml\\_value\\_u::integer](#), [MXML\\_INTEGER](#), [mxml\\_node\\_s::type](#), and [mxml\\_node\\_s::value](#).

#### 13.71.1.4 [int mxmlSetOpaque](#) ([mxml\\_node\\_t](#) \* *node*, const char \* *opaque*)

Definition at line 136 of file mxml-set.c.

References [MXML\\_OPAQUE](#), [mxml\\_value\\_u::opaque](#), [mxml\\_node\\_s::type](#), and [mxml\\_node\\_s::value](#).

#### 13.71.1.5 [int mxmlSetReal](#) ([mxml\\_node\\_t](#) \* *node*, double *real*)

Definition at line 166 of file mxml-set.c.

References [MXML\\_REAL](#), [mxml\\_value\\_u::real](#), [mxml\\_node\\_s::type](#), and [mxml\\_node\\_s::value](#).

#### 13.71.1.6 `int mxmlSetText (mxml_node_t * node, int whitespace, const char * string)`

Definition at line 193 of file mxml-set.c.

References `MXML_TEXT`, `mxml_text_s::string`, `mxml_value_u::text`, `mxml_node_s::type`, `mxml_node_s::value`, and `mxml_text_s::whitespace`.

#### 13.71.1.7 `int mxmlSetTextf (mxml_node_t * node, int whitespace, const char * format, ...)`

Definition at line 225 of file mxml-set.c.

References `mxml_strdupf()`, `MXML_TEXT`, `mxml_text_s::string`, `mxml_value_u::text`, `mxml_node_s::type`, `mxml_node_s::value`, and `mxml_text_s::whitespace`.

## 13.72 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mxml-2.2.2/mxml-string.c File Reference

```
#include "config.h"
```

### Functions

- char \* [mxml\\_strdup](#) (const char \*format, va\_list ap)
- int [mxml\\_vsnprintf](#) (char \*buffer, size\_t bufsz, const char \*format, va\_list ap)

### 13.72.1 Function Documentation

#### 13.72.1.1 char\* mxml\_strdup (const char \**format*, va\_list *ap*)

Definition at line 62 of file mxml-string.c.

References [mxml\\_vsnprintf](#)().

Referenced by [mxml\\_error](#)(), [mxmlNewTextf](#)(), and [mxmlSetTextf](#)().

#### 13.72.1.2 int mxml\_vsnprintf (char \**buffer*, size\_t *bufsize*, const char \**format*, va\_list *ap*)

Definition at line 107 of file mxml-string.c.

References [size](#).

Referenced by [mxml\\_strdup](#)().

## 13.73 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mxml-2.2.2/mxml.h File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#include <errno.h>
```

### Data Structures

- struct [mxml\\_attr\\_s](#)
- struct [mxml\\_value\\_s](#)
- struct [mxml\\_text\\_s](#)
- struct [mxml\\_custom\\_s](#)
- union [mxml\\_value\\_u](#)
- struct [mxml\\_node\\_s](#)
- struct [mxml\\_index\\_s](#)

### Defines

- #define [MXML\\_WRAP](#) 72
- #define [MXML\\_TAB](#) 8
- #define [MXML\\_NO\\_CALLBACK](#) 0
- #define [MXML\\_INTEGER\\_CALLBACK](#) mxml\_integer\_cb
- #define [MXML\\_OPAQUE\\_CALLBACK](#) mxml\_opaque\_cb
- #define [MXML\\_REAL\\_CALLBACK](#) mxml\_real\_cb
- #define [MXML\\_TEXT\\_CALLBACK](#) 0
- #define [MXML\\_NO\\_PARENT](#) 0
- #define [MXML\\_DESCEND](#) 1
- #define [MXML\\_NO\\_DESCEND](#) 0
- #define [MXML\\_DESCEND\\_FIRST](#) -1
- #define [MXML\\_WS\\_BEFORE\\_OPEN](#) 0
- #define [MXML\\_WS\\_AFTER\\_OPEN](#) 1
- #define [MXML\\_WS\\_BEFORE\\_CLOSE](#) 2
- #define [MXML\\_WS\\_AFTER\\_CLOSE](#) 3
- #define [MXML\\_ADD\\_BEFORE](#) 0
- #define [MXML\\_ADD\\_AFTER](#) 1
- #define [MXML\\_ADD\\_TO\\_PARENT](#) NULL

### Typedefs

- typedef enum [mxml\\_type\\_e](#) mxml\_type\_t
- typedef struct [mxml\\_attr\\_s](#) mxml\_attr\_t
- typedef struct [mxml\\_value\\_s](#) mxml\_element\_t
- typedef struct [mxml\\_text\\_s](#) mxml\_text\_t
- typedef struct [mxml\\_custom\\_s](#) mxml\_custom\_t

- typedef union [mxml\\_value\\_u](#) [mxml\\_value\\_t](#)
- typedef struct [mxml\\_node\\_s](#) [mxml\\_node\\_t](#)
- typedef struct [mxml\\_index\\_s](#) [mxml\\_index\\_t](#)
- typedef int(\* [mxml\\_custom\\_load\\_cb\\_t](#))([mxml\\_node\\_t](#) \*, const char \*)
- typedef char \*(\* [mxml\\_custom\\_save\\_cb\\_t](#))([mxml\\_node\\_t](#) \*)

## Enumerations

- enum [mxml\\_type\\_e](#) {  
[MXML\\_ELEMENT](#), [MXML\\_INTEGER](#), [MXML\\_OPAQUE](#), [MXML\\_REAL](#),  
[MXML\\_TEXT](#), [MXML\\_CUSTOM](#) }

## Functions

- void [mxmlAdd](#) ([mxml\\_node\\_t](#) \*parent, int where, [mxml\\_node\\_t](#) \*child, [mxml\\_node\\_t](#) \*node)
- void [mxmlDelete](#) ([mxml\\_node\\_t](#) \*node)
- const char \* [mxmlElementGetAttr](#) ([mxml\\_node\\_t](#) \*node, const char \*name)
- void [mxmlElementSetAttr](#) ([mxml\\_node\\_t](#) \*node, const char \*name, const char \*value)
- int [mxmlEntityAddCallback](#) (int(\*cb)(const char \*name))
- const char \* [mxmlEntityGetName](#) (int val)
- int [mxmlEntityGetValue](#) (const char \*name)
- void [mxmlEntityRemoveCallback](#) (int(\*cb)(const char \*name))
- [mxml\\_node\\_t](#) \* [mxmlFindElement](#) ([mxml\\_node\\_t](#) \*node, [mxml\\_node\\_t](#) \*top, const char \*name, const char \*attr, const char \*value, int descend)
- void [mxmlIndexDelete](#) ([mxml\\_index\\_t](#) \*ind)
- [mxml\\_node\\_t](#) \* [mxmlIndexEnum](#) ([mxml\\_index\\_t](#) \*ind)
- [mxml\\_node\\_t](#) \* [mxmlIndexFind](#) ([mxml\\_index\\_t](#) \*ind, const char \*element, const char \*value)
- [mxml\\_index\\_t](#) \* [mxmlIndexNew](#) ([mxml\\_node\\_t](#) \*node, const char \*element, const char \*attr)
- [mxml\\_node\\_t](#) \* [mxmlIndexReset](#) ([mxml\\_index\\_t](#) \*ind)
- [mxml\\_node\\_t](#) \* [mxmlLoadFd](#) ([mxml\\_node\\_t](#) \*top, int fd, [mxml\\_type\\_t](#)(\*cb)([mxml\\_node\\_t](#) \*))
- [mxml\\_node\\_t](#) \* [mxmlLoadFile](#) ([mxml\\_node\\_t](#) \*top, FILE \*fp, [mxml\\_type\\_t](#)(\*cb)([mxml\\_node\\_t](#) \*))
- [mxml\\_node\\_t](#) \* [mxmlLoadString](#) ([mxml\\_node\\_t](#) \*top, const char \*s, [mxml\\_type\\_t](#)(\*cb)([mxml\\_node\\_t](#) \*))
- [mxml\\_node\\_t](#) \* [mxmlNewCustom](#) ([mxml\\_node\\_t](#) \*parent, void \*data, void(\*destroy)(void \*))
- [mxml\\_node\\_t](#) \* [mxmlNewElement](#) ([mxml\\_node\\_t](#) \*parent, const char \*name)
- [mxml\\_node\\_t](#) \* [mxmlNewInteger](#) ([mxml\\_node\\_t](#) \*parent, int integer)
- [mxml\\_node\\_t](#) \* [mxmlNewOpaque](#) ([mxml\\_node\\_t](#) \*parent, const char \*opaque)
- [mxml\\_node\\_t](#) \* [mxmlNewReal](#) ([mxml\\_node\\_t](#) \*parent, double real)
- [mxml\\_node\\_t](#) \* [mxmlNewText](#) ([mxml\\_node\\_t](#) \*parent, int whitespace, const char \*string)
- [mxml\\_node\\_t](#) \* [mxmlNewTextf](#) ([mxml\\_node\\_t](#) \*parent, int whitespace, const char \*format,...)
- void [mxmlRemove](#) ([mxml\\_node\\_t](#) \*node)
- char \* [mxmlSaveAllocString](#) ([mxml\\_node\\_t](#) \*node, const char \*(\*cb)([mxml\\_node\\_t](#) \*, int))
- int [mxmlSaveFd](#) ([mxml\\_node\\_t](#) \*node, int fd, const char \*(\*cb)([mxml\\_node\\_t](#) \*, int))
- int [mxmlSaveFile](#) ([mxml\\_node\\_t](#) \*node, FILE \*fp, const char \*(\*cb)([mxml\\_node\\_t](#) \*, int))
- int [mxmlSaveString](#) ([mxml\\_node\\_t](#) \*node, char \*buffer, int bufsize, const char \*(\*cb)([mxml\\_node\\_t](#) \*, int))
- int [mxmlSetCustom](#) ([mxml\\_node\\_t](#) \*node, void \*data, void(\*destroy)(void \*))
- void [mxmlSetCustomHandlers](#) ([mxml\\_custom\\_load\\_cb\\_t](#) load, [mxml\\_custom\\_save\\_cb\\_t](#) save)
- int [mxmlSetElement](#) ([mxml\\_node\\_t](#) \*node, const char \*name)

- void [mxmlSetErrorCallback](#) (void(\*cb)(const char \*))
- int [mxmlSetInteger](#) (mxml\_node\_t \*node, int integer)
- int [mxmlSetOpaque](#) (mxml\_node\_t \*node, const char \*opaque)
- int [mxmlSetReal](#) (mxml\_node\_t \*node, double real)
- int [mxmlSetText](#) (mxml\_node\_t \*node, int whitespace, const char \*string)
- int [mxmlSetTextf](#) (mxml\_node\_t \*node, int whitespace, const char \*format,...)
- mxml\_node\_t \* [mxmlWalkNext](#) (mxml\_node\_t \*node, mxml\_node\_t \*top, int descend)
- mxml\_node\_t \* [mxmlWalkPrev](#) (mxml\_node\_t \*node, mxml\_node\_t \*top, int descend)
- void [mxml\\_error](#) (const char \*format,...)
- mxml\_type\_t [mxml\\_integer\\_cb](#) (mxml\_node\_t \*node)
- mxml\_type\_t [mxml\\_opaque\\_cb](#) (mxml\_node\_t \*node)
- mxml\_type\_t [mxml\\_real\\_cb](#) (mxml\_node\_t \*node)

### 13.73.1 Define Documentation

#### 13.73.1.1 #define MXML\_ADD\_AFTER 1

Definition at line 68 of file mxml.h.

Referenced by [add\\_variable\(\)](#), [agent\\_xml\\_compose\(\)](#), [agent\\_xml\\_compose\\_\\_agent\\_data\(\)](#), [agent\\_xml\\_compose\\_\\_create\\_row\\_nodes\(\)](#), [agent\\_xml\\_compose\\_\\_data\(\)](#), [agent\\_xml\\_compose\\_\\_gaf\\_message\(\)](#), [agent\\_xml\\_compose\\_\\_message\(\)](#), [agent\\_xml\\_compose\\_\\_mobile\\_agent\(\)](#), [agent\\_xml\\_compose\\_\\_task\(\)](#), [agent\\_xml\\_compose\\_\\_tasks\(\)](#), [fipa\\_envelope\\_Compose\(\)](#), [fipa\\_envelope\\_Compose\\_\\_envelope\(\)](#), [fipa\\_envelope\\_Compose\\_\\_params\(\)](#), [mxml\\_new\(\)](#), [mxmlAdd\(\)](#), [scan\\_file\(\)](#), and [sort\\_node\(\)](#).

#### 13.73.1.2 #define MXML\_ADD\_BEFORE 0

Definition at line 67 of file mxml.h.

Referenced by [mxmlAdd\(\)](#), [scan\\_file\(\)](#), and [sort\\_node\(\)](#).

#### 13.73.1.3 #define MXML\_ADD\_TO\_PARENT NULL

Definition at line 69 of file mxml.h.

Referenced by [add\\_variable\(\)](#), [agent\\_xml\\_compose\(\)](#), [agent\\_xml\\_compose\\_\\_create\\_row\\_nodes\(\)](#), [fipa\\_envelope\\_Compose\(\)](#), [fipa\\_envelope\\_Compose\\_\\_envelope\(\)](#), [fipa\\_envelope\\_Compose\\_\\_params\(\)](#), [mxml\\_new\(\)](#), [scan\\_file\(\)](#), and [sort\\_node\(\)](#).

#### 13.73.1.4 #define MXML\_DESCEND 1

Definition at line 58 of file mxml.h.

Referenced by [agent\\_xml\\_parse\\_\\_tasks\(\)](#), [main\(\)](#), [MC\\_LoadAgentFromFile\(\)](#), [message\\_xml\\_parse\(\)](#), [mxmlFindElement\(\)](#), [mxmlIndexNew\(\)](#), and [write\\_element\(\)](#).

#### 13.73.1.5 #define MXML\_DESCEND\_FIRST -1

Definition at line 60 of file mxml.h.

Referenced by `agent_xml_parse__fill_row_data()`, `agent_xml_parse__task()`, `agent_xml_parse__tasks()`, `fipa_envelope_HandleAclRepresentation()`, `fipa_envelope_HandleComments()`, `fipa_envelope_HandleDate()`, `fipa_envelope_HandleEnvelope()`, `fipa_envelope_HandleFrom()`, `fipa_envelope_HandleIntendedReceiver()`, `fipa_envelope_HandleParams()`, `fipa_envelope_HandlePayloadEncoding()`, `fipa_envelope_HandlePayloadLength()`, `fipa_envelope_HandleReceived()`, `fipa_envelope_HandleTo()`, `fipa_envelope_ParseAddresses()`, `fipa_envelope_ParseAgentIdentifier()`, `fipa_envelope_ParseResolvers()`, `scan_file()`, `sort_node()`, and `write_documentation()`.

#### 13.73.1.6 **#define MXML\_INTEGER\_CALLBACK mxml\_integer\_cb**

Definition at line 48 of file `mxml.h`.

Referenced by `main()`.

#### 13.73.1.7 **#define MXML\_NO\_CALLBACK 0**

Definition at line 47 of file `mxml.h`.

Referenced by `acc_connection_Thread()`, `agent_xml_compose()`, `fipa_envelope_Compose()`, `fipa_envelope_Parse()`, `main()`, `message_InitializeFromAgent()`, `message_InitializeFromConnection()`, and `message_InitializeFromString()`.

#### 13.73.1.8 **#define MXML\_NO\_DESCEND 0**

Definition at line 59 of file `mxml.h`.

Referenced by `agent_xml_parse__task()`, `agent_xml_parse__tasks()`, `fipa_envelope_HandleIntendedReceiver()`, `fipa_envelope_HandleTo()`, `fipa_envelope_ParseAddresses()`, `fipa_envelope_ParseResolvers()`, `main()`, `message_xml_parse()`, `write_documentation()`, `write_element()`, `xml_find_sibling()`, and `xml_get_deep_child()`.

#### 13.73.1.9 **#define MXML\_NO\_PARENT 0**

Definition at line 56 of file `mxml.h`.

Referenced by `agent_xml_compose__agent_code()`, `agent_xml_compose__create_row_nodes()`, `main()`, and `scan_file()`.

#### 13.73.1.10 **#define MXML\_OPAQUE\_CALLBACK mxml\_opaque\_cb**

Definition at line 50 of file `mxml.h`.

Referenced by `main()`.

#### 13.73.1.11 **#define MXML\_REAL\_CALLBACK mxml\_real\_cb**

Definition at line 52 of file `mxml.h`.

Referenced by `main()`.



**13.73.1.12 #define MXML\_TAB 8**

Definition at line 45 of file mxml.h.

Referenced by mxml\_write\_ws().

**13.73.1.13 #define MXML\_TEXT\_CALLBACK 0**

Definition at line 54 of file mxml.h.

**13.73.1.14 #define MXML\_WRAP 72**

Definition at line 44 of file mxml.h.

Referenced by mxml\_write\_node().

**13.73.1.15 #define MXML\_WS\_AFTER\_CLOSE 3**

Definition at line 65 of file mxml.h.

Referenced by mxml\_write\_node(), whitespace\_cb(), and ws\_cb().

**13.73.1.16 #define MXML\_WS\_AFTER\_OPEN 1**

Definition at line 63 of file mxml.h.

Referenced by mxml\_write\_node(), whitespace\_cb(), and ws\_cb().

**13.73.1.17 #define MXML\_WS\_BEFORE\_CLOSE 2**

Definition at line 64 of file mxml.h.

Referenced by mxml\_write\_node(), whitespace\_cb(), and ws\_cb().

**13.73.1.18 #define MXML\_WS\_BEFORE\_OPEN 0**

Definition at line 62 of file mxml.h.

Referenced by mxml\_write\_node(), whitespace\_cb(), and ws\_cb().

## **13.73.2 Typedef Documentation**

**13.73.2.1 typedef struct mxml\_attr\_s mxml\_attr\_t**

**13.73.2.2 typedef int(\* mxml\_custom\_load\_cb\_t)(mxml\_node\_t \*, const char \*)**

Definition at line 142 of file mxml.h.

**13.73.2.3 typedef char\*(\* mxml\_custom\_save\_cb\_t)(mxml\_node\_t \*)**

Definition at line 145 of file mxml.h.

**13.73.2.4** typedef struct mxml\_custom\_s mxml\_custom\_t

**13.73.2.5** typedef struct mxml\_value\_s mxml\_element\_t

**13.73.2.6** typedef struct mxml\_index\_s mxml\_index\_t

**13.73.2.7** typedef struct mxml\_node\_s mxml\_node\_t

**13.73.2.8** typedef struct mxml\_text\_s mxml\_text\_t

**13.73.2.9** typedef enum mxml\_type\_e mxml\_type\_t

**13.73.2.10** typedef union mxml\_value\_u mxml\_value\_t

### 13.73.3 Enumeration Type Documentation

**13.73.3.1** enum mxml\_type\_e

Enumerator:

*MXML\_ELEMENT*

*MXML\_INTEGER*

*MXML\_OPAQUE*

*MXML\_REAL*

*MXML\_TEXT*

*MXML\_CUSTOM*

Definition at line 76 of file mxml.h.

### 13.73.4 Function Documentation

**13.73.4.1** void mxml\_error (const char \**format*, ...)

Definition at line 49 of file mxml-private.c.

References mxml\_error\_cb, and mxml\_strdup().

Referenced by mxml\_add\_char(), mxml\_fd\_getc(), mxml\_file\_getc(), mxml\_get\_entity(), mxml\_load\_data(), mxml\_parse\_element(), mxml\_string\_getc(), mxmlElementSetAttr(), mxmlEntityAddCallback(), and mxmlIndexNew().

**13.73.4.2** mxml\_type\_t mxml\_integer\_cb (mxml\_node\_t \**node*)

Definition at line 95 of file mxml-private.c.

References MXML\_INTEGER.

**13.73.4.3** mxml\_type\_t mxml\_opaque\_cb (mxml\_node\_t \**node*)

Definition at line 108 of file mxml-private.c.

References MXML\_OPAQUE.

#### 13.73.4.4 `mxml_type_t mxml_real_cb (mxml_node_t * node)`

Definition at line 121 of file mxml-private.c.

References MXML\_REAL.

#### 13.73.4.5 `void mxmlAdd (mxml_node_t * parent, int where, mxml_node_t * child, mxml_node_t * node)`

Definition at line 62 of file mxml-node.c.

References `mxml_node_s::child`, `mxml_node_s::last_child`, `MXML_ADD_AFTER`, `MXML_ADD_BEFORE`, `mxmlRemove()`, `mxml_node_s::next`, `mxml_node_s::parent`, and `mxml_node_s::prev`.

Referenced by `add_variable()`, `agent_xml_compose()`, `agent_xml_compose__agent_data()`, `agent_xml_compose__create_row_nodes()`, `agent_xml_compose__data()`, `agent_xml_compose__gaf_message()`, `agent_xml_compose__message()`, `agent_xml_compose__mobile_agent()`, `agent_xml_compose__task()`, `agent_xml_compose__tasks()`, `fipa_envelope_Compose()`, `fipa_envelope_Compose__envelope()`, `fipa_envelope_Compose__params()`, `mxml_new()`, `scan_file()`, and `sort_node()`.

#### 13.73.4.6 `void mxmlDelete (mxml_node_t * node)`

Definition at line 196 of file mxml-node.c.

References `mxml_value_s::attrs`, `mxml_node_s::child`, `mxml_value_u::custom`, `mxml_custom_s::data`, `mxml_custom_s::destroy`, `mxml_value_u::element`, `MXML_CUSTOM`, `MXML_ELEMENT`, `MXML_INTEGER`, `MXML_OPAQUE`, `MXML_REAL`, `MXML_TEXT`, `mxmlDelete()`, `mxmlRemove()`, `mxml_attr_s::name`, `mxml_value_s::name`, `mxml_value_s::num_attrs`, `mxml_value_u::opaque`, `mxml_text_s::string`, `mxml_value_u::text`, `mxml_node_s::type`, `mxml_attr_s::value`, and `mxml_node_s::value`.

Referenced by `add_variable()`, `agent_datastate_Destroy()`, `fipa_envelope_Compose()`, `fipa_envelope_Parse()`, `main()`, `message_Destroy()`, `mxml_load_data()`, `mxmlDelete()`, `scan_file()`, and `sort_node()`.

#### 13.73.4.7 `const char* mxmlElementGetAttr (mxml_node_t * node, const char * name)`

Definition at line 43 of file mxml-attr.c.

References `mxml_value_s::attrs`, `mxml_value_u::element`, `MXML_ELEMENT`, `mxml_attr_s::name`, `mxml_value_s::num_attrs`, `mxml_node_s::type`, `mxml_attr_s::value`, and `mxml_node_s::value`.

Referenced by `agent_xml_parse__agent_code()`, `agent_xml_parse__data()`, `agent_xml_parse__task()`, `agent_xml_parse__tasks()`, `fipa_envelope_HandleReceived()`, `index_compare()`, `index_find()`, `message_xml_parse__message()`, `mxml_parse_element()`, `mxmlFindElement()`, `mxmlIndexNew()`, `scan_file()`, `sort_node()`, `type_cb()`, and `write_documentation()`.

#### 13.73.4.8 `void mxmlElementSetAttr (mxml_node_t * node, const char * name, const char * value)`

Definition at line 90 of file mxml-attr.c.

References `mxml_value_s::attrs`, `mxml_value_u::element`, `MXML_ELEMENT`, `mxml_error()`, `mxml_value_s::name`, `mxml_attr_s::name`, `mxml_value_s::num_attrs`, `mxml_node_s::type`, `mxml_attr_s::value`, and `mxml_node_s::value`.

Referenced by `add_variable()`, `agent_xml_compose__agent_code()`, `agent_xml_compose__create_row_nodes()`, `agent_xml_compose__data()`, `agent_xml_compose__message()`, `agent_xml_compose__task()`,

agent\_xml\_compose\_\_tasks(), fipa\_envelope\_Compose\_\_params(), main(), mxml\_parse\_element(), scan\_file(), sort\_node(), and update\_comment().

#### 13.73.4.9 **int mxmlEntityAddCallback (int(\*) (const char \*name) cb)**

Definition at line 65 of file mxml-entity.c.

References callbacks, mxml\_error(), and num\_callbacks.

#### 13.73.4.10 **const char\* mxmlEntityGetName (int val)**

Definition at line 91 of file mxml-entity.c.

Referenced by mxml\_write\_name(), and mxml\_write\_string().

#### 13.73.4.11 **int mxmlEntityGetValue (const char \* name)**

Definition at line 121 of file mxml-entity.c.

References callbacks, and num\_callbacks.

Referenced by mxml\_get\_entity().

#### 13.73.4.12 **void mxmlEntityRemoveCallback (int(\*) (const char \*name) cb)**

Definition at line 140 of file mxml-entity.c.

References callbacks, and num\_callbacks.

#### 13.73.4.13 **mxml\_node\_t\* mxmlFindElement (mxml\_node\_t \* node, mxml\_node\_t \* top, const char \* name, const char \* attr, const char \* value, int descend)**

Definition at line 51 of file mxml-search.c.

References mxml\_value\_u::element, MXML\_DESCEND, MXML\_ELEMENT, mxmlElementGetAttr(), mxmlWalkNext(), mxml\_value\_s::name, mxml\_node\_s::next, mxml\_node\_s::type, and mxml\_node\_s::value.

Referenced by agent\_xml\_parse\_\_fill\_row\_data(), agent\_xml\_parse\_\_task(), agent\_xml\_parse\_\_tasks(), fipa\_envelope\_HandleAclRepresentation(), fipa\_envelope\_HandleComments(), fipa\_envelope\_HandleDate(), fipa\_envelope\_HandleEnvelope(), fipa\_envelope\_HandleFrom(), fipa\_envelope\_HandleIntendedReceiver(), fipa\_envelope\_HandleParams(), fipa\_envelope\_HandlePayloadEncoding(), fipa\_envelope\_HandlePayloadLength(), fipa\_envelope\_HandleReceived(), fipa\_envelope\_HandleTo(), fipa\_envelope\_ParseAddresses(), fipa\_envelope\_ParseAgentIdentifier(), fipa\_envelope\_ParseResolvers(), main(), MC\_LoadAgentFromFile(), message\_xml\_parse(), mxmlIndexNew(), scan\_file(), sort\_node(), write\_documentation(), write\_element(), xml\_find\_sibling(), and xml\_get\_child().

#### 13.73.4.14 **void mxmlIndexDelete (mxml\_index\_t \* ind)**

Definition at line 58 of file mxml-index.c.

References mxml\_index\_s::alloc\_nodes, mxml\_index\_s::attr, and mxml\_index\_s::nodes.

Referenced by main(), and mxmlIndexNew().

**13.73.4.15 mxml\_node\_t\* mxmlIndexEnum (mxml\_index\_t \* *ind*)**

Definition at line 88 of file mxml-index.c.

References mxml\_index\_s::cur\_node, mxml\_index\_s::nodes, and mxml\_index\_s::num\_nodes.

Referenced by main(), and mxmlIndexFind().

**13.73.4.16 mxml\_node\_t\* mxmlIndexFind (mxml\_index\_t \* *ind*, const char \* *element*, const char \* *value*)**

Definition at line 118 of file mxml-index.c.

References mxml\_index\_s::attr, mxml\_index\_s::cur\_node, index\_find(), mxmlIndexEnum(), mxml\_index\_s::nodes, and mxml\_index\_s::num\_nodes.

Referenced by main().

**13.73.4.17 mxml\_index\_t\* mxmlIndexNew (mxml\_node\_t \* *node*, const char \* *element*, const char \* *attr*)**

Definition at line 301 of file mxml-index.c.

References mxml\_index\_s::alloc\_nodes, mxml\_index\_s::attr, mxml\_value\_u::element, index\_sort(), MXML\_DESCEND, mxml\_error(), mxmlElementGetAttr(), mxmlFindElement(), mxmlIndexDelete(), mxml\_value\_s::name, mxml\_index\_s::nodes, mxml\_index\_s::num\_nodes, and mxml\_node\_s::value.

Referenced by main().

**13.73.4.18 mxml\_node\_t\* mxmlIndexReset (mxml\_index\_t \* *ind*)**

Definition at line 459 of file mxml-index.c.

References mxml\_index\_s::cur\_node, mxml\_index\_s::nodes, and mxml\_index\_s::num\_nodes.

Referenced by main().

**13.73.4.19 mxml\_node\_t\* mxmlLoadFd (mxml\_node\_t \* *top*, int *fd*, mxml\_type\_t(\*) (mxml\_node\_t \*) *cb*)**

**13.73.4.20 mxml\_node\_t\* mxmlLoadFile (mxml\_node\_t \* *top*, FILE \* *fp*, mxml\_type\_t(\*) (mxml\_node\_t \*) *cb*)**

**13.73.4.21 mxml\_node\_t\* mxmlLoadString (mxml\_node\_t \* *top*, const char \* *s*, mxml\_type\_t(\*) (mxml\_node\_t \*) *cb*)**

**13.73.4.22 mxml\_node\_t\* mxmlNewCustom (mxml\_node\_t \* *parent*, void \* *data*, void(\*) (void \*) *destroy*)**

Definition at line 287 of file mxml-node.c.

References mxml\_value\_u::custom, mxml\_custom\_s::data, mxml\_custom\_s::destroy, MXML\_CUSTOM, mxml\_new(), node, and mxml\_node\_s::value.

Referenced by mxml\_load\_data().

#### 13.73.4.23 `mxmxml_node_t* mxmxmlNewElement (mxmxml_node_t * parent, const char * name)`

Definition at line 323 of file `mxmxml-node.c`.

References `mxmxml_value_u::element`, `MXMXML_ELEMENT`, `mxmxml_new()`, `mxmxml_value_s::name`, `node`, and `mxmxml_node_s::value`.

Referenced by `add_variable()`, `agent_xml_compose__agent_code()`, `agent_xml_compose__agent_data()`, `agent_xml_compose__create_row_nodes()`, `agent_xml_compose__data()`, `agent_xml_compose__gaf_message()`, `agent_xml_compose__home()`, `agent_xml_compose__message()`, `agent_xml_compose__mobile_agent()`, `agent_xml_compose__name()`, `agent_xml_compose__owner()`, `agent_xml_compose__task()`, `agent_xml_compose__tasks()`, `agent_xml_compose__wg_code()`, `fipa_envelope_Compose__acl_representation()`, `fipa_envelope_Compose__date()`, `fipa_envelope_Compose__envelope()`, `fipa_envelope_Compose__from()`, `fipa_envelope_Compose__intended_receiver()`, `fipa_envelope_Compose__params()`, `fipa_envelope_Compose__payload_encoding()`, `fipa_envelope_Compose__to()`, `main()`, `mxmxml_load_data()`, `scan_file()`, and `xml_new_cdata()`.

#### 13.73.4.24 `mxmxml_node_t* mxmxmlNewInteger (mxmxml_node_t * parent, int integer)`

Definition at line 361 of file `mxmxml-node.c`.

References `mxmxml_value_u::integer`, `MXMXML_INTEGER`, `mxmxml_new()`, `node`, and `mxmxml_node_s::value`.

Referenced by `main()`, and `mxmxml_load_data()`.

#### 13.73.4.25 `mxmxml_node_t* mxmxmlNewOpaque (mxmxml_node_t * parent, const char * opaque)`

Definition at line 392 of file `mxmxml-node.c`.

References `mxmxml_new()`, `MXMXML_OPAQUE`, `node`, `mxmxml_value_u::opaque`, and `mxmxml_node_s::value`.

Referenced by `main()`, and `mxmxml_load_data()`.

#### 13.73.4.26 `mxmxml_node_t* mxmxmlNewReal (mxmxml_node_t * parent, double real)`

Definition at line 430 of file `mxmxml-node.c`.

References `mxmxml_new()`, `MXMXML_REAL`, `node`, `mxmxml_value_u::real`, and `mxmxml_node_s::value`.

Referenced by `main()`, and `mxmxml_load_data()`.

#### 13.73.4.27 `mxmxml_node_t* mxmxmlNewText (mxmxml_node_t * parent, int whitespace, const char * string)`

Definition at line 462 of file `mxmxml-node.c`.

References `mxmxml_new()`, `MXMXML_TEXT`, `node`, `mxmxml_text_s::string`, `mxmxml_value_u::text`, `mxmxml_node_s::value`, and `mxmxml_text_s::whitespace`.

Referenced by `agent_xml_compose__create_row_nodes()`, `agent_xml_compose__home()`, `agent_xml_compose__name()`, `agent_xml_compose__owner()`, `agent_xml_compose__wg_code()`, `fipa_envelope_Compose__acl_representation()`, `fipa_envelope_Compose__date()`, `fipa_envelope_Compose__from()`, `fipa_envelope_Compose__intended_receiver()`, `fipa_envelope_Compose__payload_encoding()`, `fipa_envelope_Compose__to()`, `main()`, `mxmxml_load_data()`, and `scan_file()`.

**13.73.4.28** `mxml_node_t* mxmlNewTextf (mxml_node_t * parent, int whitespace, const char * format, ...)`

Definition at line 506 of file mxml-node.c.

References `mxml_new()`, `mxml_strdupf()`, `MXML_TEXT`, `node`, `mxml_text_s::string`, `mxml_value_u::text`, `mxml_node_s::value`, and `mxml_text_s::whitespace`.

**13.73.4.29** `void mxmlRemove (mxml_node_t * node)`

Definition at line 553 of file mxml-node.c.

References `mxml_node_s::child`, `mxml_node_s::last_child`, `mxml_node_s::next`, `mxml_node_s::parent`, and `mxml_node_s::prev`.

Referenced by `mxmlAdd()`, and `mxmlDelete()`.

**13.73.4.30** `char* mxmlSaveAllocString (mxml_node_t * node, const char *(*)(mxml_node_t *, int) cb)`

**13.73.4.31** `int mxmlSaveFd (mxml_node_t * node, int fd, const char *(*)(mxml_node_t *, int) cb)`

**13.73.4.32** `int mxmlSaveFile (mxml_node_t * node, FILE * fp, const char *(*)(mxml_node_t *, int) cb)`

**13.73.4.33** `int mxmlSaveString (mxml_node_t * node, char * buffer, int bufsize, const char *(*)(mxml_node_t *, int) cb)`

**13.73.4.34** `int mxmlSetCustom (mxml_node_t * node, void * data, void (*)(void *) destroy)`

Definition at line 46 of file mxml-set.c.

References `mxml_value_u::custom`, `mxml_custom_s::data`, `mxml_custom_s::destroy`, `MXML_CUSTOM`, `mxml_node_s::type`, and `mxml_node_s::value`.

**13.73.4.35** `void mxmlSetCustomHandlers (mxml_custom_load_cb_t load, mxml_custom_save_cb_t save)`

Definition at line 456 of file mxml-file.c.

References `mxml_custom_load_cb`, and `mxml_custom_save_cb`.

**13.73.4.36** `int mxmlSetElement (mxml_node_t * node, const char * name)`

Definition at line 79 of file mxml-set.c.

References `mxml_value_u::element`, `MXML_ELEMENT`, `mxml_value_s::name`, `mxml_node_s::type`, and `mxml_node_s::value`.

**13.73.4.37** `void mxmlSetErrorCallback (void (*)(const char *) cb)`

Definition at line 471 of file mxml-file.c.

References `mxml_error_cb`.

**13.73.4.38 int mxmlSetInteger (mxml\_node\_t \* *node*, int *integer*)**

Definition at line 109 of file mxml-set.c.

References mxml\_value\_u::integer, MXML\_INTEGER, mxml\_node\_s::type, and mxml\_node\_s::value.

**13.73.4.39 int mxmlSetOpaque (mxml\_node\_t \* *node*, const char \* *opaque*)**

Definition at line 136 of file mxml-set.c.

References MXML\_OPAQUE, mxml\_value\_u::opaque, mxml\_node\_s::type, and mxml\_node\_s::value.

**13.73.4.40 int mxmlSetReal (mxml\_node\_t \* *node*, double *real*)**

Definition at line 166 of file mxml-set.c.

References MXML\_REAL, mxml\_value\_u::real, mxml\_node\_s::type, and mxml\_node\_s::value.

**13.73.4.41 int mxmlSetText (mxml\_node\_t \* *node*, int *whitespace*, const char \* *string*)**

Definition at line 193 of file mxml-set.c.

References MXML\_TEXT, mxml\_text\_s::string, mxml\_value\_u::text, mxml\_node\_s::type, mxml\_node\_s::value, and mxml\_text\_s::whitespace.

**13.73.4.42 int mxmlSetTextf (mxml\_node\_t \* *node*, int *whitespace*, const char \* *format*, ...)**

Definition at line 225 of file mxml-set.c.

References mxml\_strdupf(), MXML\_TEXT, mxml\_text\_s::string, mxml\_value\_u::text, mxml\_node\_s::type, mxml\_node\_s::value, and mxml\_text\_s::whitespace.

**13.73.4.43 mxml\_node\_t\* mxmlWalkNext (mxml\_node\_t \* *node*, mxml\_node\_t \* *top*, int *descend*)**

Definition at line 133 of file mxml-search.c.

References mxml\_node\_s::child, mxml\_node\_s::next, and mxml\_node\_s::parent.

Referenced by mxmlFindElement(), and write\_element().

**13.73.4.44 mxml\_node\_t\* mxmlWalkPrev (mxml\_node\_t \* *node*, mxml\_node\_t \* *top*, int *descend*)**

Definition at line 169 of file mxml-search.c.

References mxml\_node\_s::last\_child, mxml\_node\_s::parent, and mxml\_node\_s::prev.



## 13.74 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mxml-2.2.2/mxmldoc.c File Reference

```
#include "config.h"
#include "mxml.h"
```

### Defines

- `#define STATE_NONE` 0
- `#define STATE_PREPROCESSOR` 1
- `#define STATE_C_COMMENT` 2
- `#define STATE_CXX_COMMENT` 3
- `#define STATE_STRING` 4
- `#define STATE_CHARACTER` 5
- `#define STATE_IDENTIFIER` 6

### Functions

- static `mxml_node_t` \* `add_variable` (`mxml_node_t` \*parent, const char \*name, `mxml_node_t` \*type)
- static void `safe_strcpy` (char \*dst, const char \*src)
- static `int` `scan_file` (const char \*filename, FILE \*fp, `mxml_node_t` \*doc)
- static void `sort_node` (`mxml_node_t` \*tree, `mxml_node_t` \*func)
- static void `update_comment` (`mxml_node_t` \*parent, `mxml_node_t` \*comment)
- static void `write_documentation` (`mxml_node_t` \*doc)
- static void `write_element` (`mxml_node_t` \*doc, `mxml_node_t` \*element)
- static void `write_string` (const char \*s)
- static const char \* `ws_cb` (`mxml_node_t` \*node, `int` where)
- `int` `main` (`int` argc, char \*argv[ ])

#### 13.74.1 Define Documentation

##### 13.74.1.1 `#define STATE_C_COMMENT` 2

Definition at line 125 of file `mxmldoc.c`.

Referenced by `scan_file()`.

##### 13.74.1.2 `#define STATE_CHARACTER` 5

Definition at line 128 of file `mxmldoc.c`.

Referenced by `scan_file()`.

##### 13.74.1.3 `#define STATE_CXX_COMMENT` 3

Definition at line 126 of file `mxmldoc.c`.

Referenced by `scan_file()`.

#### 13.74.1.4 **#define STATE\_IDENTIFIER 6**

Definition at line 129 of file mxmldoc.c.

Referenced by scan\_file().

#### 13.74.1.5 **#define STATE\_NONE 0**

Definition at line 123 of file mxmldoc.c.

Referenced by scan\_file().

#### 13.74.1.6 **#define STATE\_PREPROCESSOR 1**

Definition at line 124 of file mxmldoc.c.

Referenced by scan\_file().

#### 13.74.1.7 **#define STATE\_STRING 4**

Definition at line 127 of file mxmldoc.c.

Referenced by scan\_file().

### 13.74.2 **Function Documentation**

#### 13.74.2.1 **static mxml\_node\_t \* add\_variable (mxml\_node\_t \* *parent*, const char \* *name*, mxml\_node\_t \* *type*) [static]**

Definition at line 309 of file mxmldoc.c.

References mxml\_node\_s::child, mxml\_node\_s::last\_child, MXML\_ADD\_AFTER, MXML\_ADD\_TO\_PARENT, mxmlAdd(), mxmlDelete(), mxmlElementSetAttr(), mxmlNewElement(), mxml\_node\_s::next, node, mxml\_text\_s::string, mxml\_value\_u::text, mxml\_node\_s::value, and mxml\_text\_s::whitespace.

Referenced by scan\_file().

#### 13.74.2.2 **int main (int *argc*, char \* *argv*[])**

Definition at line 155 of file mxmldoc.c.

References MXML\_DESCEND, MXML\_NO\_CALLBACK, mxmlDelete(), mxmlElementSetAttr(), mxmlFindElement(), mxmlLoadFile(), mxmlNewElement(), mxmlSaveFile(), scan\_file(), write\_documentation(), and ws\_cb().

#### 13.74.2.3 **static void safe\_strcpy (char \* *dst*, const char \* *src*) [static]**

Definition at line 423 of file mxmldoc.c.

Referenced by update\_comment().

**13.74.2.4 static int scan\_file (const char \*filename, FILE \*fp, mxml\_node\_t \* doc) [static]**

Definition at line 438 of file mxmldoc.c.

References add\_variable(), mxml\_node\_s::child, mxml\_value\_u::element, mxml\_node\_s::last\_child, MXML\_ADD\_AFTER, MXML\_ADD\_BEFORE, MXML\_ADD\_TO\_PARENT, MXML\_DESCEND\_FIRST, MXML\_NO\_PARENT, mxmlAdd(), mxmlDelete(), mxmlElementGetAttr(), mxmlElementSetAttr(), mxmlFindElement(), mxmlNewElement(), mxmlNewText(), mxml\_value\_s::name, mxml\_node\_s::next, node, sort\_node(), STATE\_C\_COMMENT, STATE\_CHARACTER, STATE\_CXX\_COMMENT, STATE\_IDENTIFIER, STATE\_NONE, STATE\_PREPROCESSOR, STATE\_STRING, mxml\_text\_s::string, mxml\_value\_u::text, update\_comment(), mxml\_node\_s::value, and mxml\_text\_s::whitespace.

Referenced by main().

**13.74.2.5 static void sort\_node (mxml\_node\_t \* tree, mxml\_node\_t \* func) [static]**

Definition at line 1613 of file mxmldoc.c.

References mxml\_node\_s::child, mxml\_value\_u::element, MXML\_ADD\_AFTER, MXML\_ADD\_BEFORE, MXML\_ADD\_TO\_PARENT, MXML\_DESCEND\_FIRST, mxmlAdd(), mxmlDelete(), mxmlElementGetAttr(), mxmlElementSetAttr(), mxmlFindElement(), mxml\_value\_s::name, mxml\_node\_s::next, mxml\_node\_s::parent, and mxml\_node\_s::value.

Referenced by scan\_file().

**13.74.2.6 static void update\_comment (mxml\_node\_t \* parent, mxml\_node\_t \* comment) [static]**

Definition at line 1701 of file mxmldoc.c.

References mxml\_value\_u::element, mxmlElementSetAttr(), mxml\_value\_s::name, safe\_strcpy(), mxml\_text\_s::string, mxml\_value\_u::text, and mxml\_node\_s::value.

Referenced by scan\_file().

**13.74.2.7 static void write\_documentation (mxml\_node\_t \* doc) [static]**

Definition at line 1801 of file mxmldoc.c.

References mxml\_node\_s::child, MXML\_DESCEND\_FIRST, MXML\_NO\_DESCEND, MXML\_VERSION, mxmlElementGetAttr(), mxmlFindElement(), and write\_element().

Referenced by main().

**13.74.2.8 static void write\_element (mxml\_node\_t \* doc, mxml\_node\_t \* element) [static]**

Definition at line 2611 of file mxmldoc.c.

References mxml\_node\_s::child, MXML\_DESCEND, MXML\_NO\_DESCEND, MXML\_TEXT, mxmlFindElement(), mxmlWalkNext(), node, mxml\_text\_s::string, mxml\_value\_u::text, mxml\_node\_s::type, mxml\_node\_s::value, mxml\_text\_s::whitespace, and write\_string().

Referenced by write\_documentation().

**13.74.2.9 static void write\_string (const char \* *s*) [static]**

Definition at line 2656 of file mxmldoc.c.

Referenced by write\_element().

**13.74.2.10 static const char \* ws\_cb (mxml\_node\_t \* *node*, int *where*) [static]**

Definition at line 2714 of file mxmldoc.c.

References mxml\_value\_u::element, MXML\_WS\_AFTER\_CLOSE, MXML\_WS\_AFTER\_OPEN, MXML\_WS\_BEFORE\_CLOSE, MXML\_WS\_BEFORE\_OPEN, mxml\_value\_s::name, mxml\_node\_s::parent, and mxml\_node\_s::value.

Referenced by main().

## 13.75 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mxml-2.2.2/test/class.cxx File Reference

### Data Structures

- class [foo\\_c](#)

### Variables

- [foo\\_c f](#)
- [foo\\_c foo](#)
- [bar](#) = b

### 13.75.1 Variable Documentation

#### 13.75.1.1 [bar](#) = b

Definition at line 77 of file class.cxx.

#### 13.75.1.2 [foo\\_c f](#)

Referenced by [foo\\_float\\_function\(\)](#), [main\(\)](#), [MC\\_Initialize\(\)](#), [md5\\_check\(\)](#), [md5\\_file\(\)](#), [read\\_encrypted\\_file\(\)](#), [sha1\\_check\(\)](#), [sha1\\_file\(\)](#), [sha2\\_check\(\)](#), [sha2\\_file\(\)](#), [sha4\\_file\(\)](#), [x509parse\\_certfile\(\)](#), and [x509parse\\_keyfile\(\)](#).

#### 13.75.1.3 [foo\\_c foo](#)

## 13.76 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mxml-2.2.2/test/enum.cxx File Reference

### Typedefs

- typedef enum [foo\\_enum\\_e](#) [foo\\_enum\\_t](#)

### Enumerations

- enum [foo\\_enum\\_e](#) { [FOO\\_ONE](#), [FOO\\_TWO](#), [FOO\\_RED](#), [FOO\\_BLUE](#) }

#### 13.76.1 Typedef Documentation

##### 13.76.1.1 typedef enum [foo\\_enum\\_e](#) [foo\\_enum\\_t](#)

#### 13.76.2 Enumeration Type Documentation

##### 13.76.2.1 enum [foo\\_enum\\_e](#)

Enumerator:

*FOO\_ONE*  
*FOO\_TWO*  
*FOO\_RED*  
*FOO\_BLUE*

Definition at line 1 of file enum.cxx.

## 13.77 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mxml-2.2.2/test/function.cxx File Reference

### Functions

- void [foo\\_void\\_function](#) ([int](#) *one*, float \**two*, const char \**three*)
- float [foo\\_float\\_function](#) ([int](#) *one*, const char \**two*)
- [int](#) [foo\\_default\\_string](#) ([int](#) *one*, const char \**two*="2")
- [int](#) [foo\\_default\\_int](#) ([int](#) *one*, [int](#) *two*=2)

### 13.77.1 Function Documentation

#### 13.77.1.1 [int](#) [foo\\_default\\_int](#) ([int](#) *one*, [int](#) *two* = 2)

Definition at line 65 of file `function.cxx`.

#### 13.77.1.2 [int](#) [foo\\_default\\_string](#) ([int](#) *one*, const char \* *two* = "2")

Definition at line 45 of file `function.cxx`.

#### 13.77.1.3 [float](#) [foo\\_float\\_function](#) ([int](#) *one*, const char \* *two*)

Definition at line 26 of file `function.cxx`.

References [f](#).

#### 13.77.1.4 [void](#) [foo\\_void\\_function](#) ([int](#) *one*, float \* *two*, const char \* *three*)

Definition at line 6 of file `function.cxx`.

## 13.78 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mxml-2.2.2/test/struct.cxx File Reference

### Data Structures

- struct [foo\\_s](#)

### Typedefs

- typedef struct [foo\\_s](#) [foo\\_t](#)

### 13.78.1 Typedef Documentation

#### 13.78.1.1 typedef struct [foo\\_s](#) [foo\\_t](#)



## 13.79 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/mxml-2.2.2/testmxml.c File Reference

```
#include "config.h"
#include "mxml.h"
#include <unistd.h>
#include <fcntl.h>
```

### Defines

- `#define O_BINARY 0`

### Functions

- `mxml_type_t type_cb (mxml_node_t *node)`
- `const char * whitespace_cb (mxml_node_t *node, int where)`
- `int main (int argc, char *argv[])`

#### 13.79.1 Define Documentation

##### 13.79.1.1 `#define O_BINARY 0`

Definition at line 42 of file testmxml.c.

Referenced by `main()`.

#### 13.79.2 Function Documentation

##### 13.79.2.1 `int main (int argc, char *argv[])`

Definition at line 59 of file testmxml.c.

References `mxml_node_s::child`, `mxml_value_u::element`, `f`, `mxml_value_u::integer`, `mxml_node_s::last_child`, `MXML_DESCEND`, `MXML_ELEMENT`, `MXML_INTEGER`, `MXML_INTEGER_CALLBACK`, `MXML_NO_CALLBACK`, `MXML_NO_DESCEND`, `MXML_NO_PARENT`, `MXML_OPAQUE`, `MXML_OPAQUE_CALLBACK`, `MXML_REAL`, `MXML_REAL_CALLBACK`, `MXML_TEXT`, `mxmlDelete()`, `mxmlFindElement()`, `mxmlIndexDelete()`, `mxmlIndexEnum()`, `mxmlIndexFind()`, `mxmlIndexNew()`, `mxmlIndexReset()`, `mxmlLoadFd()`, `mxmlLoadFile()`, `mxmlLoadString()`, `mxmlNewElement()`, `mxmlNewInteger()`, `mxmlNewOpaque()`, `mxmlNewReal()`, `mxmlNewText()`, `mxmlSaveFd()`, `mxmlSaveFile()`, `mxmlSaveString()`, `mxml_value_s::name`, `mxml_node_s::next`, `node`, `mxml_index_s::num_nodes`, `O_BINARY`, `mxml_value_u::opaque`, `mxml_value_u::real`, `mxml_text_s::string`, `mxml_value_u::text`, `mxml_node_s::type`, `type_cb()`, `mxml_node_s::value`, `mxml_text_s::whitespace`, and `whitespace_cb()`.

##### 13.79.2.2 `mxml_type_t type_cb (mxml_node_t *node)`

Definition at line 542 of file testmxml.c.

References `mxml_value_u::element`, `MXML_INTEGER`, `MXML_OPAQUE`, `MXML_REAL`, `MXML_TEXT`, `mxmlElementGetAttr()`, `mxml_value_s::name`, and `mxml_node_s::value`.

Referenced by `main()`.

### 13.79.2.3 `const char* whitespace_cb (mxml_node_t * node, int where)`

Definition at line 571 of file `testmxml.c`.

References `mxml_node_s::child`, `mxml_value_u::element`, `MXML_WS_AFTER_CLOSE`, `MXML_WS_AFTER_OPEN`, `MXML_WS_BEFORE_CLOSE`, `MXML_WS_BEFORE_OPEN`, `mxml_value_s::name`, `mxml_node_s::parent`, and `mxml_node_s::value`.

## 13.80 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/asm.c File Reference

```
#include "../include/mc_platform.h"
#include "../include/message.h"
#include "asm.h"
#include "asm_message_composer.h"
#include "config.h"
#include "mc_dh.h"
```

### **13.81   /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/asm.h File Reference**

```
#include "config.h"
#include "../include/ap_queue_template.h"
#include "../include/data_structures.h"
#include "../mc_list/list.h"
#include "asm_node.h"
```

## 13.82 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/asm\_message\_composer.c File Reference

```
#include "asm_message_composer.h"  
#include "config.h"  
#include "../include/mc_platform.h"
```

### 13.83 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/asm\_message\_composer.h File Reference

```
#include <mxml.h>
#include "asm.h"
#include "config.h"
```

## **13.84 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/asm\_message\_parser.c File Reference**

```
#include "asm_message_parser.h"  
#include "../include/xml_helper.h"  
#include "config.h"
```

### **13.85   /home/dko/Projects/mobilec/tags/MobileC- v1.10.10/src/security/asm\_message\_parser.h File Reference**

```
#include "../include/mc_error.h"
#include "../include/xml_parser.h"
#include "asm_node.h"
#include "config.h"
```



## **13.86 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/asm\_node.c File Reference**

```
#include <netdb.h>
#include "config.h"
#include "asm.h"
#include "asm_message_parser.h"
#include "asm_node.h"
#include "xyssl-0.9/include/xyssl/havege.h"
#include "xyssl-0.9/include/xyssl/bignum.h"
```

### 13.87 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/asm\_node.h File Reference

```
#include <mxml.h>
#include <netinet/in.h>
#include "../include/macros.h"
#include "../include/message.h"
#include "xyssl-0.9/include/xyssl/dhm.h"
#include "xyssl-0.9/include/xyssl/rsa.h"
#include "xyssl-0.9/include/xyssl/aes.h"
#include "config.h"
```

## 13.88 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/interface.c File Reference

```
#include "interface.h"
```

### Functions

- static void [mystrncpy\\_binary](#) (char \*dest, char \*src, [int](#) start\_index, [int](#) length)
- static void [separate\\_key\\_parts](#) (char \*key, char \*N, char \*E, char \*D, char \*P, char \*Q, char \*DP, char \*DQ, char \*QP)
- [int](#) [rsa\\_encryption](#) (char \*publickey, char \*plaintext, char \*ciphertext)
- [int](#) [rsa\\_decryption](#) (char \*ciphertext, char \*plaintext, char \*privatekey)
- static [int](#) [append\\_nonce\\_to\\_MA](#) ([int](#) \*my\_nonce, char \*MA\_file)
- static [int](#) [remove\\_nonce\\_from\\_MA](#) (char \*MA\_file)
- static [int](#) [extract\\_nonce\\_from\\_MA](#) ([int](#) sockfd, [int](#) \*my\_nonce, char \*MA\_file)
- [int](#) [read\\_known\\_host\\_file](#) (char \*pubkey, char \*hname, char \*filename)
- [int](#) [read\\_encrypted\\_file](#) (char \*enfile, char \*string, unsigned char \*passphrase)
- [int](#) [initiate\\_migration\\_process](#) ([int](#) new\_fd, [int](#) \*my\_nonce, char \*pubkey, char \*privkey, unsigned char \*aes\_key)
- [int](#) [reply\\_migration\\_process](#) ([int](#) sockfd, [int](#) \*my\_nonce, char \*pubkey, char \*privkey, unsigned char \*aes\_key)
- void [generate\\_AES\\_key](#) (char \*key)
- [int](#) [aes\\_en\\_de](#) ([int](#) mode, char \*infile, char \*outfile, unsigned char \*AES\_key, [int](#) \*nonce, [int](#) new\_fd)
- [int](#) [send\\_AES\\_en\\_MA](#) ([int](#) sockfd, [int](#) \*my\_nonce, char \*outfile, char \*pubkey)
- [int](#) [receive\\_AES\\_en\\_MA](#) ([int](#) new\_fd, [int](#) \*nonce, char \*infile, char \*privkey)
- [int](#) [generate\\_RSA\\_keys\\_plaintext](#) (char \*pubkeyfile, char \*privkeyfile)
- [int](#) [generate\\_RSA\\_keys\\_ciphertext](#) (char \*pubkeyfile, char \*privkeyfile, unsigned char \*passphrase)

### 13.88.1 Function Documentation

#### 13.88.1.1 [int](#) [aes\\_en\\_de](#) ([int](#) mode, char \*infile, char \*outfile, unsigned char \*AES\_key, [int](#) \*nonce, [int](#) new\_fd)

Definition at line 1112 of file interface.c.

References [aes\\_crypt\\_ecb\(\)](#), [AES\\_DECRYPT](#), [AES\\_ENCRYPT](#), [aes\\_setkey\\_dec\(\)](#), [aes\\_setkey\\_enc\(\)](#), [append\\_nonce\\_to\\_MA\(\)](#), [extract\\_nonce\\_from\\_MA\(\)](#), [int](#), [MODE\\_DECRYPT](#), [MODE\\_ENCRYPT](#), [sha2\\_finish\(\)](#), [sha2\\_hmac\\_finish\(\)](#), [sha2\\_hmac\\_starts\(\)](#), [sha2\\_hmac\\_update\(\)](#), [sha2\\_starts\(\)](#), and [sha2\\_update\(\)](#).

Referenced by [auth\\_rece\\_send\\_msg\(\)](#), [generate\\_RSA\\_keys\\_ciphertext\(\)](#), [read\\_encrypted\\_file\(\)](#), and [rece\\_de\\_msg\(\)](#).

#### 13.88.1.2 static [int](#) [append\\_nonce\\_to\\_MA](#) ([int](#) \*my\_nonce, char \*MA\_file) [static]

Definition at line 176 of file interface.c.

Referenced by [aes\\_en\\_de\(\)](#).

**13.88.1.3 static int extract\_nonce\_from\_MA (int *sockfd*, int \* *my\_nonce*, char \* *MA\_file*)  
[static]**

Definition at line 253 of file interface.c.

References `remove_nonce_from_MA()`, and `send`.

Referenced by `aes_en_de()`.

**13.88.1.4 void generate\_AES\_key (char \* *key*)**

Definition at line 1089 of file interface.c.

References `havege_init()`, and `havege_rand()`.

Referenced by `initiate_migration_process()`, and `reply_migration_process()`.

**13.88.1.5 int generate\_RSA\_keys\_ciphertext (char \* *pubkeyfile*, char \* *privkeyfile*, unsigned char \* *passphrase*)**

Definition at line 1743 of file interface.c.

References `aes_en_de()`, `rsa_context::D`, `rsa_context::DP`, `rsa_context::DQ`, `rsa_context::E`, `EXPONENT`, `havege_init()`, `havege_rand()`, `KEY_SIZE`, `mpi_write_file()`, `rsa_context::N`, `rsa_context::P`, `rsa_context::Q`, `rsa_context::QP`, `rsa_gen_key()`, `rsa_init()`, and `RSA_PKCS_V15`.

Referenced by `main()`.

**13.88.1.6 int generate\_RSA\_keys\_plaintext (char \* *pubkeyfile*, char \* *privkeyfile*)**

Definition at line 1684 of file interface.c.

References `rsa_context::D`, `rsa_context::DP`, `rsa_context::DQ`, `rsa_context::E`, `EXPONENT`, `havege_init()`, `havege_rand()`, `KEY_SIZE`, `mpi_write_file()`, `rsa_context::N`, `rsa_context::P`, `rsa_context::Q`, `rsa_context::QP`, `rsa_gen_key()`, `rsa_init()`, and `RSA_PKCS_V15`.

Referenced by `main()`.

**13.88.1.7 int initiate\_migration\_process (int *new\_fd*, int \* *my\_nonce*, char \* *pubkey*, char \* *privkey*, unsigned char \* *aes\_key*)**

Definition at line 422 of file interface.c.

References `generate_AES_key()`, `havege_init()`, `havege_rand()`, `md5()`, `rsa_decryption()`, `rsa_encryption()`, and `send`.

Referenced by `auth_rece_send_msg()`.

**13.88.1.8 static void mystrncpy\_binary (char \* *dest*, char \* *src*, int *start\_index*, int *length*)  
[static]**

Definition at line 13 of file interface.c.

**13.88.1.9 int read\_encrypted\_file (char \* *enfile*, char \* *string*, unsigned char \* *passphrase*)**

Definition at line 388 of file interface.c.

References aes\_en\_de(), and f.

Referenced by MC\_Initialize().

**13.88.1.10 int read\_known\_host\_file (char \* *pubkey*, char \* *hname*, char \* *filename*)**

Definition at line 319 of file interface.c.

Referenced by auth\_conn\_rece\_key(), and auth\_rece\_send\_msg().

**13.88.1.11 int receive\_AES\_en\_MA (int *new\_fd*, int \* *nonce*, char \* *infile*, char \* *privkey*)**

Definition at line 1546 of file interface.c.

References rsa\_decryption(), and size.

Referenced by rece\_de\_msg().

**13.88.1.12 static int remove\_nonce\_from\_MA (char \* *MA\_file*) [static]**

Definition at line 209 of file interface.c.

References int.

Referenced by extract\_nonce\_from\_MA().

**13.88.1.13 int reply\_migration\_process (int *sockfd*, int \* *my\_nonce*, char \* *pubkey*, char \* *privkey*, unsigned char \* *aes\_key*)**

Definition at line 767 of file interface.c.

References generate\_AES\_key(), md5(), rsa\_decryption(), rsa\_encryption(), and send.

Referenced by auth\_conn\_rece\_key().

**13.88.1.14 int rsa\_decryption (char \* *ciphertext*, char \* *plaintext*, char \* *privatekey*)**

Definition at line 125 of file interface.c.

References rsa\_context::D, rsa\_context::DP, rsa\_context::DQ, rsa\_context::E, rsa\_context::len, mpi\_msb(), mpi\_read\_mystring(), rsa\_context::N, rsa\_context::P, P, rsa\_context::Q, rsa\_context::QP, rsa\_check\_privkey(), RSA\_DE, rsa\_init(), rsa\_pkcs1\_decrypt(), RSA\_PKCS\_V15, and separate\_key\_parts().

Referenced by initiate\_migration\_process(), receive\_AES\_en\_MA(), and reply\_migration\_process().

**13.88.1.15 int rsa\_encryption (char \* *publickey*, char \* *plaintext*, char \* *ciphertext*)**

Definition at line 88 of file interface.c.

References rsa\_context::E, rsa\_context::len, mpi\_msb(), mpi\_read\_mystring(), rsa\_context::N, rsa\_check\_pubkey(), RSA\_EN, rsa\_init(), rsa\_pkcs1\_encrypt(), RSA\_PKCS\_V15, and separate\_key\_parts().

Referenced by `initiate_migration_process()`, `reply_migration_process()`, and `send_AES_en_MA()`.

**13.88.1.16** `int send_AES_en_MA (int sockfd, int * my_nonce, char * outfile, char * pubkey)`

Definition at line 1410 of file `interface.c`.

References `int`, `rsa_encryption()`, `send`, and `size`.

Referenced by `auth_rece_send_msg()`.

**13.88.1.17** `static void separate_key_parts (char * key, char * N, char * E, char * D, char * P,  
char * Q, char * DP, char * DQ, char * QP) [static]`

Definition at line 25 of file `interface.c`.

Referenced by `rsa_decryption()`, and `rsa_encryption()`.

## 13.89 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/interface.h File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/stat.h>
#include "xyssl-0.9/include/xyssl/havege.h"
#include "xyssl-0.9/include/xyssl/bignum.h"
#include "xyssl-0.9/include/xyssl/rsa.h"
#include "xyssl-0.9/include/xyssl/sha2.h"
#include "xyssl-0.9/include/xyssl/sha1.h"
#include "xyssl-0.9/include/xyssl/aes.h"
#include "xyssl-0.9/include/xyssl/md5.h"
```

### Defines

- #define [PATH](#) xyssl-0.9/include/xyssl
- #define [MODE\\_ENCRYPT](#) 0
- #define [MODE\\_DECRYPT](#) 1
- #define [MAXDATASIZE](#) 4096
- #define [KEY\\_SIZE](#) 1024
- #define [EXPONENT](#) 65537
- #define [RSA\\_EN](#) 0
- #define [RSA\\_DE](#) 1

### Functions

- static void [mystrncpy](#) (char \*dest, char \*src, [int](#) start\_index, [int](#) length)
- [int](#) [read\\_known\\_host\\_file](#) (char \*pubkey, char \*hname, char \*filename)
- static void [separate\\_key\\_parts](#) (char \*pubkey, char \*N, char \*E, char \*D, char \*P, char \*Q, char \*DP, char \*DQ, char \*QP)
- [int](#) [read\\_encrypted\\_file](#) (char \*enfile, char \*string, unsigned char \*passphrase)
- [int](#) [rsa\\_encryption](#) (char \*publickey, char \*plaintext, char \*ciphertext)
- [int](#) [rsa\\_decryption](#) (char \*ciphertext, char \*plaintext, char \*privatekeyfile)
- [int](#) [initiate\\_migration\\_process](#) ([int](#) sockfd, [int](#) \*nonce, char \*publickey, char \*privatekey, unsigned char \*aes\_key)
- [int](#) [reply\\_migration\\_process](#) ([int](#) new\_fd, [int](#) \*nonce, char \*publickey, char \*privatekey, unsigned char \*aes\_key)
- static void [generate\\_AES\\_key](#) (char \*key)
- [int](#) [generate\\_encrypt\\_send\\_AES\\_key](#) ([int](#) sockfd, [int](#) \*nonce, unsigned char \*key, char \*publickey, char \*privkey)
- [int](#) [receive\\_decrypt\\_AES\\_key](#) ([int](#) new\_fd, [int](#) \*nonce, unsigned char \*key, char \*privkey, char \*publickey)
- [int](#) [aes\\_en\\_de](#) ([int](#) mode, char \*infile, char \*outfile, unsigned char \*AES\_key, [int](#) \*nonce, [int](#) sockfd)

- static [int append\\_nonce\\_to\\_MA](#) ([int](#) \*my\_nonce, char \*MA\_file)
- static [int extract\\_nonce\\_from\\_MA](#) ([int](#) sockfd, [int](#) \*my\_nonce, char \*MA\_file)
- [int send\\_AES\\_en\\_MA](#) ([int](#) sockfd, [int](#) \*nonce, char \*outfile, char \*peer\_pubkey)
- [int receive\\_AES\\_en\\_MA](#) ([int](#) new\_fd, [int](#) \*nonce, char \*infile, char \*privatekey)
- [int receiving\\_verifying\\_MA\\_RSA](#) ([int](#) sockfd, char \*privkeyfile)
- [int receiving\\_decrypting\\_MA\\_RSA](#) ([int](#) new\_fd, char \*privkeyfile)
- [int generate\\_RSA\\_keys\\_plaintext](#) (char \*pubkeyfile, char \*privkeyfile)
- [int generate\\_RSA\\_keys\\_ciphertext](#) (char \*pubkeyfile, char \*privkeyfile, unsigned char \*passphrase)

### 13.89.1 Define Documentation

#### 13.89.1.1 #define EXPONENT 65537

Definition at line 25 of file interface.h.

Referenced by [generate\\_RSA\\_keys\\_ciphertext\(\)](#), [generate\\_RSA\\_keys\\_plaintext\(\)](#), and [main\(\)](#).

#### 13.89.1.2 #define KEY\_SIZE 1024

Definition at line 24 of file interface.h.

Referenced by [generate\\_RSA\\_keys\\_ciphertext\(\)](#), [generate\\_RSA\\_keys\\_plaintext\(\)](#), and [main\(\)](#).

#### 13.89.1.3 #define MAXDATASIZE 4096

Definition at line 22 of file interface.h.

#### 13.89.1.4 #define MODE\_DECRYPT 1

Definition at line 21 of file interface.h.

Referenced by [aes\\_en\\_de\(\)](#), and [main\(\)](#).

#### 13.89.1.5 #define MODE\_ENCRYPT 0

Definition at line 20 of file interface.h.

Referenced by [aes\\_en\\_de\(\)](#), and [main\(\)](#).

#### 13.89.1.6 #define PATH xyssl-0.9/include/xyssl

Definition at line 9 of file interface.h.

#### 13.89.1.7 #define RSA\_DE 1

Definition at line 28 of file interface.h.

Referenced by [rsa\\_decryption\(\)](#).



### 13.89.1.8 #define RSA\_EN 0

Definition at line 27 of file interface.h.

Referenced by rsa\_encryption().

## 13.89.2 Function Documentation

### 13.89.2.1 int aes\_en\_de(int mode, char \*infile, char \*outfile, unsigned char \*AES\_key, int \*nonce, int sockfd)

Definition at line 1112 of file interface.c.

References aes\_crypt\_ecb(), AES\_DECRYPT, AES\_ENCRYPT, aes\_setkey\_dec(), aes\_setkey\_enc(), append\_nonce\_to\_MA(), extract\_nonce\_from\_MA(), int, MODE\_DECRYPT, MODE\_ENCRYPT, sha2\_finish(), sha2\_hmac\_finish(), sha2\_hmac\_starts(), sha2\_hmac\_update(), sha2\_starts(), and sha2\_update().

Referenced by auth\_rece\_send\_msg(), generate\_RSA\_keys\_ciphertext(), read\_encrypted\_file(), and rece\_de\_msg().

### 13.89.2.2 static int append\_nonce\_to\_MA(int \*my\_nonce, char \*MA\_file) [static]

### 13.89.2.3 static int extract\_nonce\_from\_MA(int sockfd, int \*my\_nonce, char \*MA\_file) [static]

### 13.89.2.4 static void generate\_AES\_key(char \*key) [static]

Definition at line 1089 of file interface.c.

References havege\_init(), and havege\_rand().

Referenced by initiate\_migration\_process(), and reply\_migration\_process().

### 13.89.2.5 int generate\_encrypt\_send\_AES\_key(int sockfd, int \*nonce, unsigned char \*key, char \*publickey, char \*privkey)

### 13.89.2.6 int generate\_RSA\_keys\_ciphertext(char \*pubkeyfile, char \*privkeyfile, unsigned char \*passphrase)

Definition at line 1743 of file interface.c.

References aes\_en\_de(), rsa\_context::D, rsa\_context::DP, rsa\_context::DQ, rsa\_context::E, EXPONENT, havege\_init(), havege\_rand(), KEY\_SIZE, mpi\_write\_file(), rsa\_context::N, rsa\_context::P, rsa\_context::Q, rsa\_context::QP, rsa\_gen\_key(), rsa\_init(), and RSA\_PKCS\_V15.

Referenced by main().

### 13.89.2.7 int generate\_RSA\_keys\_plaintext(char \*pubkeyfile, char \*privkeyfile)

Definition at line 1684 of file interface.c.

References rsa\_context::D, rsa\_context::DP, rsa\_context::DQ, rsa\_context::E, EXPONENT, havege\_init(), havege\_rand(), KEY\_SIZE, mpi\_write\_file(), rsa\_context::N, rsa\_context::P, rsa\_context::Q, rsa\_context::QP, rsa\_gen\_key(), rsa\_init(), and RSA\_PKCS\_V15.

Referenced by `main()`.

**13.89.2.8** `int initiate_migration_process (int sockfd, int * nonce, char * publickey, char * privatekey, unsigned char * aes_key)`

Definition at line 422 of file `interface.c`.

References `generate_AES_key()`, `havege_init()`, `havege_rand()`, `md5()`, `rsa_decryption()`, `rsa_encryption()`, and `send`.

Referenced by `auth_rece_send_msg()`.

**13.89.2.9** `static void mystrncpy (char * dest, char * src, int start_index, int length) [static]`

**13.89.2.10** `int read_encrypted_file (char * enfile, char * string, unsigned char * passphrase)`

Definition at line 388 of file `interface.c`.

References `aes_en_de()`, and `f`.

Referenced by `MC_Initialize()`.

**13.89.2.11** `int read_known_host_file (char * pubkey, char * hname, char * filename)`

Definition at line 319 of file `interface.c`.

Referenced by `auth_conn_rece_key()`, and `auth_rece_send_msg()`.

**13.89.2.12** `int receive_AES_en_MA (int new_fd, int * nonce, char * infile, char * privatekey)`

Definition at line 1546 of file `interface.c`.

References `rsa_decryption()`, and `size`.

Referenced by `rece_de_msg()`.

**13.89.2.13** `int receive_decrypt_AES_key (int new_fd, int * nonce, unsigned char * key, char * privkey, char * publickey)`

**13.89.2.14** `int receiving_decrypting_MA_RSA (int new_fd, char * privkeyfile)`

**13.89.2.15** `int receiving_verifying_MA_RSA (int sockfd, char * privkeyfile)`

**13.89.2.16** `int reply_migration_process (int new_fd, int * nonce, char * publickey, char * privatekey, unsigned char * aes_key)`

Definition at line 767 of file `interface.c`.

References `generate_AES_key()`, `md5()`, `rsa_decryption()`, `rsa_encryption()`, and `send`.

Referenced by `auth_conn_rece_key()`.

**13.89.2.17** `int rsa_decryption (char * ciphertext, char * plaintext, char * privatekeyfile)`

Definition at line 125 of file interface.c.

References `rsa_context::D`, `rsa_context::DP`, `rsa_context::DQ`, `rsa_context::E`, `rsa_context::len`, `mpi_msb()`, `mpi_read_mystring()`, `rsa_context::N`, `rsa_context::P`, `P`, `rsa_context::Q`, `rsa_context::QP`, `rsa_check_privkey()`, `RSA_DE`, `rsa_init()`, `rsa_pkcs1_decrypt()`, `RSA_PKCS_V15`, and `separate_key_parts()`.

Referenced by `initiate_migration_process()`, `receive_AES_en_MA()`, and `reply_migration_process()`.

**13.89.2.18** `int rsa_encryption (char * publickey, char * plaintext, char * ciphertext)`

Definition at line 88 of file interface.c.

References `rsa_context::E`, `rsa_context::len`, `mpi_msb()`, `mpi_read_mystring()`, `rsa_context::N`, `rsa_check_pubkey()`, `RSA_EN`, `rsa_init()`, `rsa_pkcs1_encrypt()`, `RSA_PKCS_V15`, and `separate_key_parts()`.

Referenced by `initiate_migration_process()`, `reply_migration_process()`, and `send_AES_en_MA()`.

**13.89.2.19** `int send_AES_en_MA (int sockfd, int * nonce, char * outfile, char * peer_pubkey)`

Definition at line 1410 of file interface.c.

References `int`, `rsa_encryption()`, `send`, and `size`.

Referenced by `auth_rece_send_msg()`.

**13.89.2.20** `static void separate_key_parts (char * pubkey, char * N, char * E, char * D, char * P, char * Q, char * DP, char * DQ, char * QP) [static]`

### 13.90 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/mc\_dh.c File Reference

```
#include <stdio.h>
#include "../include/mc_error.h"
#include "xyssl-0.9/include/xyssl/dhm.h"
#include "xyssl-0.9/include/xyssl/havege.h"
#include "mc_dh.h"
#include "asm_node.h"
#include "config.h"
```

## **13.91 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/mc\_dh.h File Reference**

```
#include "xyssl-0.9/include/xyssl/rsa.h"
#include "asm_node.h"
#include "config.h"
```

## 13.92 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/aes.h File Reference

### Data Structures

- struct [aes\\_context](#)  
*AES context structure.*

### Defines

- #define [AES\\_ENCRYPT](#) 1
- #define [AES\\_DECRYPT](#) 0

### Functions

- void [aes\\_setkey\\_enc](#) ([aes\\_context](#) \*ctx, unsigned char \*key, [int](#) keysize)  
*AES key schedule (encryption).*
- void [aes\\_setkey\\_dec](#) ([aes\\_context](#) \*ctx, unsigned char \*key, [int](#) keysize)  
*AES key schedule (decryption).*
- void [aes\\_crypt\\_ecb](#) ([aes\\_context](#) \*ctx, [int](#) mode, unsigned char input[16], unsigned char output[16])  
*AES-ECB block encryption/decryption.*
- void [aes\\_crypt\\_cbc](#) ([aes\\_context](#) \*ctx, [int](#) mode, [int](#) length, unsigned char iv[16], unsigned char \*input, unsigned char \*output)  
*AES-CBC buffer encryption/decryption.*
- void [aes\\_crypt\\_cfb](#) ([aes\\_context](#) \*ctx, [int](#) mode, [int](#) length, [int](#) \*iv\_off, unsigned char iv[16], unsigned char \*input, unsigned char \*output)  
*AES-CFB buffer encryption/decryption.*
- [int](#) [aes\\_self\\_test](#) ([int](#) verbose)  
*Checkup routine.*

### 13.92.1 Detailed Description

Definition in file [aes.h](#).

### 13.92.2 Define Documentation

#### 13.92.2.1 #define AES\_DECRYPT 0

Definition at line 8 of file [aes.h](#).

Referenced by `aes_crypt_cbc()`, `aes_crypt_cfb()`, `aes_crypt_ecb()`, `aes_en_de()`, `aes_self_test()`, `main()`, and `ssl_decrypt_buf()`.

### 13.92.2.2 `#define AES_ENCRYPT 1`

Definition at line 7 of file `aes.h`.

Referenced by `aes_crypt_cfb()`, `aes_en_de()`, `main()`, and `ssl_encrypt_buf()`.

## 13.92.3 Function Documentation

### 13.92.3.1 `void aes_crypt_cbc (aes_context * ctx, int mode, int length, unsigned char iv[16], unsigned char * input, unsigned char * output)`

AES-CBC buffer encryption/decryption.

#### Parameters:

*ctx* AES context  
*mode* AES\_ENCRYPT or AES\_DECRYPT  
*length* length of the input data  
*iv* initialization vector (updated after use)  
*input* buffer holding the input data  
*output* buffer holding the output data

Definition at line 732 of file `aes.c`.

References `aes_crypt_ecb()`, and `AES_DECRYPT`.

Referenced by `aes_self_test()`, `main()`, `ssl_decrypt_buf()`, and `ssl_encrypt_buf()`.

### 13.92.3.2 `void aes_crypt_cfb (aes_context * ctx, int mode, int length, int * iv_off, unsigned char iv[16], unsigned char * input, unsigned char * output)`

AES-CFB buffer encryption/decryption.

#### Parameters:

*ctx* AES context  
*mode* AES\_ENCRYPT or AES\_DECRYPT  
*length* length of the input data  
*iv\_off* offset in IV (updated after use)  
*iv* initialization vector (updated after use)  
*input* buffer holding the input data  
*output* buffer holding the output data

Definition at line 787 of file `aes.c`.

References `aes_crypt_ecb()`, `AES_DECRYPT`, and `AES_ENCRYPT`.

Referenced by `aes_self_test()`.

### 13.92.3.3 void aes\_crypt\_ecb (aes\_context \* *ctx*, int *mode*, unsigned char *input*[16], unsigned char *output*[16])

AES-ECB block encryption/decryption.

#### Parameters:

*ctx* AES context  
*mode* AES\_ENCRYPT or AES\_DECRYPT  
*input* 16-byte input block  
*output* 16-byte output block

Definition at line 639 of file aes.c.

References AES\_DECRYPT, AES\_FROUND, AES\_RROUND, FSb, GET\_ULONG\_LE, aes\_context::nr, PUT\_ULONG\_LE, aes\_context::rk, and RSb.

Referenced by aes\_crypt\_cbc(), aes\_crypt\_cfb(), aes\_en\_de(), aes\_self\_test(), and main().

### 13.92.3.4 int aes\_self\_test (int *verbose*)

Checkup routine.

#### Returns:

0 if successful, or 1 if the test failed

Definition at line 902 of file aes.c.

References aes\_crypt\_cbc(), aes\_crypt\_cfb(), aes\_crypt\_ecb(), AES\_DECRYPT, aes\_setkey\_dec(), aes\_setkey\_enc(), aes\_test\_cbc\_dec, aes\_test\_cbc\_enc, aes\_test\_cfb\_dec, aes\_test\_cfb\_enc, aes\_test\_ecb\_dec, aes\_test\_ecb\_enc, buf, and prv.

Referenced by main().

### 13.92.3.5 void aes\_setkey\_dec (aes\_context \* *ctx*, unsigned char \* *key*, int *keysize*)

AES key schedule (decryption).

#### Parameters:

*ctx* AES context to be initialized  
*key* decryption key  
*keysize* must be 128, 192 or 256

Definition at line 542 of file aes.c.

References aes\_setkey\_enc(), aes\_context::buf, FSb, aes\_context::nr, aes\_context::rk, RT0, RT1, RT2, and RT3.

Referenced by aes\_en\_de(), aes\_self\_test(), main(), and ssl\_derive\_keys().

### 13.92.3.6 void aes\_setkey\_enc (aes\_context \* *ctx*, unsigned char \* *key*, int *keysize*)

AES key schedule (encryption).



**Parameters:**

*ctx* AES context to be initialized

*key* encryption key

*keysize* must be 128, 192 or 256

Definition at line 439 of file aes.c.

References `aes_gen_tables()`, `aes_init_done`, `aes_context::buf`, `FSb`, `GET_ULONG_LE`, `aes_context::nr`, `RCON`, and `aes_context::rk`.

Referenced by `aes_en_de()`, `aes_self_test()`, `aes_setkey_dec()`, `main()`, and `ssl_derive_keys()`.

## 13.93 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/arc4.h File Reference

### Data Structures

- struct [arc4\\_context](#)  
*ARC4 context structure.*

### Functions

- void [arc4\\_setup](#) ([arc4\\_context](#) \*ctx, unsigned char \*key, [int](#) keylen)  
*ARC4 key schedule.*
- void [arc4\\_crypt](#) ([arc4\\_context](#) \*ctx, unsigned char \*buf, [int](#) buflen)  
*ARC4 cipher function.*
- [int](#) [arc4\\_self\\_test](#) ([int](#) verbose)

### 13.93.1 Detailed Description

Definition in file [arc4.h](#).

### 13.93.2 Function Documentation

#### 13.93.2.1 void [arc4\\_crypt](#) ([arc4\\_context](#) \* ctx, unsigned char \* buf, [int](#) buflen)

ARC4 cipher function.

##### Parameters:

*ctx* ARC4 context  
*buf* buffer to be processed  
*buflen* amount of data in buf

Definition at line 63 of file [arc4.c](#).

References [arc4\\_context::m](#), [arc4\\_context::x](#), and [arc4\\_context::y](#).

Referenced by [arc4\\_self\\_test\(\)](#), [main\(\)](#), [ssl\\_decrypt\\_buf\(\)](#), and [ssl\\_encrypt\\_buf\(\)](#).

#### 13.93.2.2 [int](#) [arc4\\_self\\_test](#) ([int](#) verbose)

Definition at line 122 of file [arc4.c](#).

References [arc4\\_crypt\(\)](#), [arc4\\_setup\(\)](#), [arc4\\_test\\_ct](#), [arc4\\_test\\_key](#), [arc4\\_test\\_pt](#), and [buf](#).

Referenced by [main\(\)](#).

### 13.93.2.3 void arc4\_setup (arc4\_context \* *ctx*, unsigned char \* *key*, int *keylen*)

ARC4 key schedule.

#### Parameters:

*ctx* ARC4 context to be initialized

*key* the secret key

*keylen* length of the key

Definition at line 35 of file arc4.c.

References arc4\_context::m, arc4\_context::x, and arc4\_context::y.

Referenced by arc4\_self\_test(), main(), and ssl\_derive\_keys().

## 13.94 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/base64.h File Reference

### Defines

- `#define XYSSL_ERR_BASE64_BUFFER_TOO_SMALL -0x0010`
- `#define XYSSL_ERR_BASE64_INVALID_CHARACTER -0x0012`

### Functions

- `int base64_encode` (unsigned char \*dst, int \*dlen, unsigned char \*src, int slen)  
*Encode a buffer into base64 format.*
- `int base64_decode` (unsigned char \*dst, int \*dlen, unsigned char \*src, int slen)  
*Decode a base64-formatted buffer.*
- `int base64_self_test` (int verbose)  
*Checkup routine.*

### 13.94.1 Detailed Description

Definition in file [base64.h](#).

### 13.94.2 Define Documentation

#### 13.94.2.1 `#define XYSSL_ERR_BASE64_BUFFER_TOO_SMALL -0x0010`

Definition at line 7 of file [base64.h](#).

Referenced by [base64\\_decode\(\)](#), and [base64\\_encode\(\)](#).

#### 13.94.2.2 `#define XYSSL_ERR_BASE64_INVALID_CHARACTER -0x0012`

Definition at line 8 of file [base64.h](#).

Referenced by [base64\\_decode\(\)](#), [x509parse\\_crt\(\)](#), and [x509parse\\_key\(\)](#).

### 13.94.3 Function Documentation

#### 13.94.3.1 `int base64_decode` (unsigned char \* dst, int \* dlen, unsigned char \* src, int slen)

Decode a base64-formatted buffer.

#### Parameters:

- dst* destination buffer
- dlen* size of the buffer

## 13.94

/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/base64.h File Reference 649

---

*src* source buffer

*slen* amount of data to be decoded

### Returns:

0 if successful, XYSSL\_ERR\_BASE64\_BUFFER\_TOO\_SMALL, or XYSSL\_ERR\_BASE64\_INVALID\_DATA if the input data is not correct. \*dlen is always updated to reflect the amount of data that has (or would have) been written.

### Note:

Call this function with \*dlen = 0 to obtain the required buffer size in \*dlen

Definition at line 121 of file base64.c.

References base64\_dec\_map, XYSSL\_ERR\_BASE64\_BUFFER\_TOO\_SMALL, and XYSSL\_ERR\_BASE64\_INVALID\_CHARACTER.

Referenced by base64\_self\_test(), x509parse\_crt(), and x509parse\_key().

### 13.94.3.2 int base64\_encode (unsigned char \*dst, int \*dlen, unsigned char \*src, int slen)

Encode a buffer into base64 format.

#### Parameters:

*dst* destination buffer

*dlen* size of the buffer

*src* source buffer

*slen* amount of data to be encoded

### Returns:

0 if successful, or XYSSL\_ERR\_BASE64\_BUFFER\_TOO\_SMALL. \*dlen is always updated to reflect the amount of data that has (or would have) been written.

### Note:

Call this function with \*dlen = 0 to obtain the required buffer size in \*dlen

Definition at line 58 of file base64.c.

References base64\_enc\_map, and XYSSL\_ERR\_BASE64\_BUFFER\_TOO\_SMALL.

Referenced by base64\_self\_test().

### 13.94.3.3 int base64\_self\_test (int verbose)

Checkup routine.

### Returns:

0 if successful, or 1 if the test failed

Definition at line 206 of file base64.c.

References base64\_decode(), base64\_encode(), base64\_test\_dec, and base64\_test\_enc.

Referenced by main().

## 13.95 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/bignum.h File Reference

```
#include <stdio.h>
```

### Data Structures

- struct [mpi](#)  
*MPI structure.*

### Defines

- #define [XYSSL\\_ERR\\_MPI\\_FILE\\_IO\\_ERROR](#) -0x0002
- #define [XYSSL\\_ERR\\_MPI\\_BAD\\_INPUT\\_DATA](#) -0x0004
- #define [XYSSL\\_ERR\\_MPI\\_INVALID\\_CHARACTER](#) -0x0006
- #define [XYSSL\\_ERR\\_MPI\\_BUFFER\\_TOO\\_SMALL](#) -0x0008
- #define [XYSSL\\_ERR\\_MPI\\_NEGATIVE\\_VALUE](#) -0x000A
- #define [XYSSL\\_ERR\\_MPI\\_DIVISION\\_BY\\_ZERO](#) -0x000C
- #define [XYSSL\\_ERR\\_MPI\\_NOT\\_ACCEPTABLE](#) -0x000E
- #define [MPI\\_CHK](#)(f) if( ( ret = f ) != 0 ) goto cleanup

### Typedefs

- typedef unsigned long [t\\_int](#)
- typedef unsigned long long [t\\_dbl](#)

### Functions

- void [mpi\\_init](#) ([mpi](#) \*X,...)  
*Initialize one or more [mpi](#).*
- void [mpi\\_free](#) ([mpi](#) \*X,...)  
*Unallocate one or more [mpi](#).*
- int [mpi\\_grow](#) ([mpi](#) \*X, int nlimbs)  
*Enlarge to the specified number of limbs.*
- int [mpi\\_copy](#) ([mpi](#) \*X, [mpi](#) \*Y)  
*Copy the contents of Y into X.*
- void [mpi\\_swap](#) ([mpi](#) \*X, [mpi](#) \*Y)  
*Swap the contents of X and Y.*
- int [mpi\\_lset](#) ([mpi](#) \*X, int z)  
*Set value from integer.*

- `int mpi_lsb (mpi *X)`  
*Return the number of least significant bits.*
- `int mpi_msb (mpi *X)`  
*Return the number of most significant bits.*
- `int mpi_size (mpi *X)`  
*Return the total size in bytes.*
- `int mpi_read_string (mpi *X, int radix, char *s)`  
*Import from an ASCII string.*
- `int mpi_write_string (mpi *X, int radix, char *s, int *slen)`  
*Export into an ASCII string.*
- `int mpi_read_file (mpi *X, int radix, FILE *fin)`  
*Read X from an opened file.*
- `int mpi_read_mystring (mpi *X, int radix, char *s)`
- `int mpi_write_file (char *p, mpi *X, int radix, FILE *fout)`  
*Write X into an opened file, or stdout.*
- `int mpi_read_binary (mpi *X, unsigned char *buf, int buflen)`  
*Import X from unsigned binary data, big endian.*
- `int mpi_write_binary (mpi *X, unsigned char *buf, int buflen)`  
*Export X into unsigned binary data, big endian.*
- `int mpi_shift_l (mpi *X, int count)`  
*Left-shift:  $X \ll = \text{count}$ .*
- `int mpi_shift_r (mpi *X, int count)`  
*Right-shift:  $X \gg = \text{count}$ .*
- `int mpi_cmp_abs (mpi *X, mpi *Y)`  
*Compare unsigned values.*
- `int mpi_cmp_mpi (mpi *X, mpi *Y)`  
*Compare signed values.*
- `int mpi_cmp_int (mpi *X, int z)`  
*Compare signed values.*
- `int mpi_add_abs (mpi *X, mpi *A, mpi *B)`  
*Unsigned addition:  $X = |A| + |B|$ .*
- `int mpi_sub_abs (mpi *X, mpi *A, mpi *B)`  
*Unsigned subtraction:  $X = |A| - |B|$ .*

- `int mpi_add_mpi (mpi *X, mpi *A, mpi *B)`  
*Signed addition:  $X = A + B$ .*
- `int mpi_sub_mpi (mpi *X, mpi *A, mpi *B)`  
*Signed subtraction:  $X = A - B$ .*
- `int mpi_add_int (mpi *X, mpi *A, int b)`  
*Signed addition:  $X = A + b$ .*
- `int mpi_sub_int (mpi *X, mpi *A, int b)`  
*Signed subtraction:  $X = A - b$ .*
- `int mpi_mul_mpi (mpi *X, mpi *A, mpi *B)`  
*Baseline multiplication:  $X = A * B$ .*
- `int mpi_mul_int (mpi *X, mpi *A, t_int b)`  
*Baseline multiplication:  $X = A * b$ .*
- `int mpi_div_mpi (mpi *Q, mpi *R, mpi *A, mpi *B)`  
*Division by mpi:  $A = Q * B + R$ .*
- `int mpi_div_int (mpi *Q, mpi *R, mpi *A, int b)`  
*Division by int:  $A = Q * b + R$ .*
- `int mpi_mod_mpi (mpi *R, mpi *A, mpi *B)`  
*Modulo:  $R = A \bmod B$ .*
- `int mpi_mod_int (t_int *r, mpi *A, int b)`  
*Modulo:  $r = A \bmod b$ .*
- `int mpi_exp_mod (mpi *X, mpi *A, mpi *E, mpi *N, mpi *_RR)`  
*Sliding-window exponentiation:  $X = A^E \bmod N$ .*
- `int mpi_gcd (mpi *G, mpi *A, mpi *B)`  
*Greatest common divisor:  $G = \gcd(A, B)$ .*
- `int mpi_inv_mod (mpi *X, mpi *A, mpi *N)`  
*Modular inverse:  $X = A^{-1} \bmod N$ .*
- `int mpi_is_prime (mpi *X, int(*f_rng)(void *), void *p_rng)`  
*Miller-Rabin primality test.*
- `int mpi_gen_prime (mpi *X, int nbits, int dh_flag, int(*f_rng)(void *), void *p_rng)`  
*Prime number generation.*
- `int mpi_self_test (int verbose)`  
*Checkup routine.*



**13.95.1 Detailed Description**

---

Definition in file [bignum.h](#).

**13.95.2 Define Documentation****13.95.2.1 #define MPI\_CHK(f) if( ( ret = f ) != 0 ) goto cleanup**

Definition at line 17 of file bignum.h.

Referenced by dhm\_calc\_secret(), dhm\_make\_params(), dhm\_make\_public(), mpi\_add\_abs(), mpi\_add\_mpi(), mpi\_copy(), mpi\_div\_mpi(), mpi\_exp\_mod(), mpi\_gcd(), mpi\_gen\_prime(), mpi\_inv\_mod(), mpi\_is\_prime(), mpi\_lset(), mpi\_mod\_mpi(), mpi\_mul\_mpi(), mpi\_read\_binary(), mpi\_read\_string(), mpi\_self\_test(), mpi\_shift\_l(), mpi\_sub\_abs(), mpi\_sub\_mpi(), mpi\_write\_file(), mpi\_write\_hlp(), mpi\_write\_string(), rsa\_check\_privkey(), rsa\_gen\_key(), rsa\_private(), and rsa\_public().

**13.95.2.2 #define XYSSL\_ERR\_MPI\_BAD\_INPUT\_DATA -0x0004**

Definition at line 10 of file bignum.h.

Referenced by mpi\_exp\_mod(), mpi\_gen\_prime(), mpi\_inv\_mod(), mpi\_read\_string(), mpi\_write\_hlp(), and mpi\_write\_string().

**13.95.2.3 #define XYSSL\_ERR\_MPI\_BUFFER\_TOO\_SMALL -0x0008**

Definition at line 12 of file bignum.h.

Referenced by mpi\_write\_binary(), and mpi\_write\_string().

**13.95.2.4 #define XYSSL\_ERR\_MPI\_DIVISION\_BY\_ZERO -0x000C**

Definition at line 14 of file bignum.h.

Referenced by mpi\_div\_mpi(), and mpi\_mod\_int().

**13.95.2.5 #define XYSSL\_ERR\_MPI\_FILE\_IO\_ERROR -0x0002**

Definition at line 9 of file bignum.h.

Referenced by mpi\_read\_file(), and mpi\_write\_file().

**13.95.2.6 #define XYSSL\_ERR\_MPI\_INVALID\_CHARACTER -0x0006**

Definition at line 11 of file bignum.h.

Referenced by mpi\_get\_digit().

**13.95.2.7 #define XYSSL\_ERR\_MPI\_NEGATIVE\_VALUE -0x000A**

Definition at line 13 of file bignum.h.

Referenced by mpi\_sub\_abs().

### 13.95.2.8 `#define XYSSL_ERR_MPI_NOT_ACCEPTABLE -0x000E`

Definition at line 15 of file `bignum.h`.

Referenced by `mpi_gen_prime()`, `mpi_inv_mod()`, and `mpi_is_prime()`.

## 13.95.3 Typedef Documentation

### 13.95.3.1 `typedef unsigned long long t_dbl`

Definition at line 39 of file `bignum.h`.

### 13.95.3.2 `typedef unsigned long t_int`

Definition at line 30 of file `bignum.h`.

## 13.95.4 Function Documentation

### 13.95.4.1 `int mpi_add_abs (mpi * X, mpi * A, mpi * B)`

Unsigned addition:  $X = |A| + |B|$ .

#### Returns:

0 if successful, 1 if memory allocation failed

Definition at line 694 of file `bignum.c`.

References `MPI_CHK`, `mpi_copy()`, `mpi_grow()`, `mpi::n`, and `mpi::p`.

Referenced by `mpi_add_mpi()`, and `mpi_sub_mpi()`.

### 13.95.4.2 `int mpi_add_int (mpi * X, mpi * A, int b)`

Signed addition:  $X = A + b$ .

#### Returns:

0 if successful, 1 if memory allocation failed

Definition at line 860 of file `bignum.c`.

References `mpi_add_mpi()`, `mpi::n`, `mpi::p`, and `mpi::s`.

Referenced by `mpi_gen_prime()`, and `mpi_read_string()`.

### 13.95.4.3 `int mpi_add_mpi (mpi * X, mpi * A, mpi * B)`

Signed addition:  $X = A + B$ .

#### Returns:

0 if successful, 1 if memory allocation failed

### 13.95

/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/bignum.h

#### File Reference

655

Definition at line 798 of file bignum.c.

References `mpi_add_abs()`, `MPI_CHK`, `mpi_cmp_abs()`, `mpi_sub_abs()`, and `mpi::s`.

Referenced by `mpi_add_int()`, `mpi_div_mpi()`, `mpi_inv_mod()`, `mpi_mod_mpi()`, and `rsa_private()`.

#### 13.95.4.4 `int mpi_cmp_abs (mpi * X, mpi * Y)`

Compare unsigned values.

##### Returns:

1 if  $|X|$  is greater than  $|Y|$ , -1 if  $|X|$  is lesser than  $|Y|$  or 0 if  $|X|$  is equal to  $|Y|$

Definition at line 615 of file bignum.c.

References `mpi::n`, and `mpi::p`.

Referenced by `mpi_add_mpi()`, `mpi_div_mpi()`, `mpi_montmul()`, `mpi_sub_abs()`, and `mpi_sub_mpi()`.

#### 13.95.4.5 `int mpi_cmp_int (mpi * X, int z)`

Compare signed values.

##### Returns:

1 if  $X$  is greater than  $z$ , -1 if  $X$  is lesser than  $z$  or 0 if  $X$  is equal to  $z$

Definition at line 678 of file bignum.c.

References `mpi_cmp_mpi()`, `mpi::n`, `mpi::p`, and `mpi::s`.

Referenced by `mpi_div_mpi()`, `mpi_exp_mod()`, `mpi_gcd()`, `mpi_inv_mod()`, `mpi_is_prime()`, `mpi_mod_mpi()`, `mpi_write_hlp()`, `rsa_check_privkey()`, and `rsa_gen_key()`.

#### 13.95.4.6 `int mpi_cmp_mpi (mpi * X, mpi * Y)`

Compare signed values.

##### Returns:

1 if  $X$  is greater than  $Y$ , -1 if  $X$  is lesser than  $Y$  or 0 if  $X$  is equal to  $Y$

Definition at line 645 of file bignum.c.

References `mpi::n`, `mpi::p`, and `mpi::s`.

Referenced by `dhm_make_params()`, `dhm_make_public()`, `mpi_cmp_int()`, `mpi_div_mpi()`, `mpi_exp_mod()`, `mpi_gcd()`, `mpi_inv_mod()`, `mpi_is_prime()`, `mpi_mod_mpi()`, `mpi_self_test()`, `rsa_check_privkey()`, `rsa_gen_key()`, `rsa_private()`, and `rsa_public()`.

#### 13.95.4.7 `int mpi_copy (mpi * X, mpi * Y)`

Copy the contents of  $Y$  into  $X$ .

##### Returns:

0 if successful, 1 if memory allocation failed

Definition at line 128 of file bignum.c.

References `ciL`, `MPI_CHK`, `mpi_grow()`, `mpi::n`, `mpi::p`, and `mpi::s`.

Referenced by `mpi_add_abs()`, `mpi_div_mpi()`, `mpi_exp_mod()`, `mpi_gcd()`, `mpi_inv_mod()`, `mpi_is_prime()`, `mpi_mul_mpi()`, `mpi_sub_abs()`, and `mpi_write_string()`.

#### 13.95.4.8 `int mpi_div_int (mpi * Q, mpi * R, mpi * A, int b)`

Division by int:  $A = Q * b + R$ .

##### Returns:

0 if successful, 1 if memory allocation failed, `XYSSL_ERR_MPI_DIVISION_BY_ZERO` if  $b == 0$

##### Note:

Either `Q` or `R` can be `NULL`.

Definition at line 1173 of file bignum.c.

References `mpi_div_mpi()`, `mpi::n`, `mpi::p`, and `mpi::s`.

Referenced by `main()`, and `mpi_write_hlp()`.

#### 13.95.4.9 `int mpi_div_mpi (mpi * Q, mpi * R, mpi * A, mpi * B)`

Division by `mpi`:  $A = Q * B + R$ .

##### Returns:

0 if successful, 1 if memory allocation failed, `XYSSL_ERR_MPI_DIVISION_BY_ZERO` if  $B == 0$

##### Note:

Either `Q` or `R` can be `NULL`.

Definition at line 1008 of file bignum.c.

References `biH`, `biL`, `mpi_add_mpi()`, `MPI_CHK`, `mpi_cmp_abs()`, `mpi_cmp_int()`, `mpi_cmp_mpi()`, `mpi_copy()`, `mpi_free()`, `mpi_grow()`, `mpi_init()`, `mpi_lset()`, `mpi_msb()`, `mpi_mul_int()`, `mpi_shift_l()`, `mpi_shift_r()`, `mpi_sub_mpi()`, `mpi::n`, `mpi::p`, `mpi::s`, and `XYSSL_ERR_MPI_DIVISION_BY_ZERO`.

Referenced by `mpi_div_int()`, `mpi_mod_mpi()`, and `mpi_self_test()`.

#### 13.95.4.10 `int mpi_exp_mod (mpi * X, mpi * A, mpi * E, mpi * N, mpi * _RR)`

Sliding-window exponentiation:  $X = A^E \bmod N$ .

##### Returns:

0 if successful, 1 if memory allocation failed, `XYSSL_ERR_MPI_BAD_INPUT_DATA` if  $N$  is negative or even

##### Note:

`_RR` is used to avoid re-computing  $R * R \bmod N$  across multiple calls, which speeds up things a bit. It can be set to `NULL` if the extra performance is unneeded.

### 13.95

/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/bignum.h

#### File Reference

657

Definition at line 1328 of file bignum.c.

References `biL`, `MPI_CHK`, `mpi_cmp_int()`, `mpi_cmp_mpi()`, `mpi_copy()`, `mpi_free()`, `mpi_grow()`, `mpi_init()`, `mpi_lset()`, `mpi_mod_mpi()`, `mpi_montg_init()`, `mpi_montmul()`, `mpi_montred()`, `mpi_msb()`, `mpi_shift_l()`, `mpi::n`, `mpi::p`, and `XYSSL_ERR_MPI_BAD_INPUT_DATA`.

Referenced by `dhm_calc_secret()`, `dhm_make_params()`, `dhm_make_public()`, `main()`, `mpi_is_prime()`, `mpi_self_test()`, `rsa_private()`, and `rsa_public()`.

#### 13.95.4.11 void mpi\_free (mpi \* X, ...)

Unallocate one or more [mpi](#).

Definition at line 73 of file bignum.c.

References `ciL`, `mpi::n`, `mpi::p`, and `mpi::s`.

Referenced by `dhm_free()`, `main()`, `mpi_div_mpi()`, `mpi_exp_mod()`, `mpi_gcd()`, `mpi_gen_prime()`, `mpi_inv_mod()`, `mpi_is_prime()`, `mpi_mul_mpi()`, `mpi_read_string()`, `mpi_self_test()`, `mpi_sub_abs()`, `mpi_write_string()`, `rsa_check_privkey()`, `rsa_free()`, `rsa_gen_key()`, `rsa_private()`, and `rsa_public()`.

#### 13.95.4.12 int mpi\_gcd (mpi \* G, mpi \* A, mpi \* B)

Greatest common divisor:  $G = \gcd(A, B)$ .

##### Returns:

0 if successful, 1 if memory allocation failed

Definition at line 1507 of file bignum.c.

References `MPI_CHK`, `mpi_cmp_int()`, `mpi_cmp_mpi()`, `mpi_copy()`, `mpi_free()`, `mpi_init()`, `mpi_lset()`, `mpi_mul_mpi()`, `mpi_shift_r()`, `mpi_sub_abs()`, `mpi::p`, and `mpi::s`.

Referenced by `mpi_inv_mod()`, `rsa_check_privkey()`, and `rsa_gen_key()`.

#### 13.95.4.13 int mpi\_gen\_prime (mpi \* X, int nbits, int dh\_flag, int(\*) (void \*) f\_rng, void \* p\_rng)

Prime number generation.

##### Parameters:

*X* destination [mpi](#)

*nbits* required size of *X* in bits

*dh\_flag* if 1, then  $(X-1)/2$  will be prime too

*f\_rng* RNG function

*p\_rng* RNG parameter

##### Returns:

0 if successful (probably prime), 1 if memory allocation failed, `XYSSL_ERR_MPI_BAD_INPUT_DATA` if *nbits* is  $< 3$

Definition at line 1778 of file bignum.c.

References BITS\_TO\_LIMBS, ciL, mpi\_add\_int(), MPI\_CHK, mpi\_free(), mpi\_grow(), mpi\_init(), mpi\_is\_prime(), mpi\_lset(), mpi\_msb(), mpi\_shift\_l(), mpi\_shift\_r(), mpi\_sub\_int(), mpi::p, XYSSL\_ERR\_MPI\_BAD\_INPUT\_DATA, and XYSSL\_ERR\_MPI\_NOT\_ACCEPTABLE.

Referenced by main(), and rsa\_gen\_key().

#### 13.95.4.14 int mpi\_grow (mpi \* X, int nblimbs)

Enlarge to the specified number of limbs.

##### Returns:

0 if successful, 1 if memory allocation failed

Definition at line 100 of file bignum.c.

References ciL, mpi::n, and mpi::p.

Referenced by dhm\_make\_params(), dhm\_make\_public(), mpi\_add\_abs(), mpi\_copy(), mpi\_div\_mpi(), mpi\_exp\_mod(), mpi\_gen\_prime(), mpi\_is\_prime(), mpi\_lset(), mpi\_mul\_mpi(), mpi\_read\_binary(), mpi\_read\_string(), and mpi\_shift\_l().

#### 13.95.4.15 void mpi\_init (mpi \* X, ...)

Initialize one or more [mpi](#).

Definition at line 52 of file bignum.c.

References mpi::n, mpi::p, and mpi::s.

Referenced by main(), mpi\_div\_mpi(), mpi\_exp\_mod(), mpi\_gcd(), mpi\_gen\_prime(), mpi\_inv\_mod(), mpi\_is\_prime(), mpi\_mul\_mpi(), mpi\_read\_string(), mpi\_self\_test(), mpi\_sub\_abs(), mpi\_write\_string(), rsa\_check\_privkey(), rsa\_gen\_key(), rsa\_private(), and rsa\_public().

#### 13.95.4.16 int mpi\_inv\_mod (mpi \* X, mpi \* A, mpi \* N)

Modular inverse:  $X = A^{-1} \bmod N$ .

##### Returns:

0 if successful, 1 if memory allocation failed, XYSSL\_ERR\_MPI\_BAD\_INPUT\_DATA if N is negative or nil XYSSL\_ERR\_MPI\_NOT\_ACCEPTABLE if A has no inverse mod N

Definition at line 1549 of file bignum.c.

References mpi\_add\_mpi(), MPI\_CHK, mpi\_cmp\_int(), mpi\_cmp\_mpi(), mpi\_copy(), mpi\_free(), mpi\_gcd(), mpi\_init(), mpi\_lset(), mpi\_mod\_mpi(), mpi\_shift\_r(), mpi\_sub\_mpi(), mpi::p, XYSSL\_ERR\_MPI\_BAD\_INPUT\_DATA, and XYSSL\_ERR\_MPI\_NOT\_ACCEPTABLE.

Referenced by main(), mpi\_self\_test(), and rsa\_gen\_key().

#### 13.95.4.17 int mpi\_is\_prime (mpi \* X, int(\*) (void \*) f\_rng, void \* p\_rng)

Miller-Rabin primality test.

**Returns:**

0 if successful (probably prime), 1 if memory allocation failed, XYSSL\_ERR\_MPI\_NOT\_ACCEPTABLE if X is not prime

Definition at line 1667 of file bignum.c.

References ciL, MPI\_CHK, mpi\_cmp\_int(), mpi\_cmp\_mpi(), mpi\_copy(), mpi\_exp\_mod(), mpi\_free(), mpi\_grow(), mpi\_init(), mpi\_lsb(), mpi\_mod\_int(), mpi\_mod\_mpi(), mpi\_msb(), mpi\_mul\_mpi(), mpi\_shift\_r(), mpi\_sub\_int(), mpi::n, mpi::p, R, mpi::s, small\_prime, and XYSSL\_ERR\_MPI\_NOT\_ACCEPTABLE.

Referenced by main(), and mpi\_gen\_prime().

**13.95.4.18 int mpi\_lsb (mpi \* X)**

Return the number of least significant bits.

Definition at line 185 of file bignum.c.

References biL, int, mpi::n, and mpi::p.

Referenced by mpi\_is\_prime().

**13.95.4.19 int mpi\_lset (mpi \* X, int z)**

Set value from integer.

**Returns:**

0 if successful, 1 if memory allocation failed

Definition at line 167 of file bignum.c.

References ciL, MPI\_CHK, mpi\_grow(), mpi::n, mpi::p, and mpi::s.

Referenced by dhm\_make\_params(), dhm\_make\_public(), mpi\_div\_mpi(), mpi\_exp\_mod(), mpi\_gcd(), mpi\_gen\_prime(), mpi\_inv\_mod(), mpi\_mul\_mpi(), mpi\_read\_binary(), mpi\_read\_string(), and rsa\_gen\_key().

**13.95.4.20 int mpi\_mod\_int (t\_int \* r, mpi \* A, int b)**

Modulo:  $r = A \bmod b$ .

**Returns:**

0 if successful, 1 if memory allocation failed, XYSSL\_ERR\_MPI\_DIVISION\_BY\_ZERO if  $b == 0$

Definition at line 1209 of file bignum.c.

References biH, mpi::n, mpi::p, and XYSSL\_ERR\_MPI\_DIVISION\_BY\_ZERO.

Referenced by mpi\_is\_prime(), and mpi\_write\_hlp().

**13.95.4.21 int mpi\_mod\_mpi (mpi \* R, mpi \* A, mpi \* B)**

Modulo:  $R = A \bmod B$ .

**Returns:**

0 if successful, 1 if memory allocation failed, `XYSSL_ERR_MPI_DIVISION_BY_ZERO` if  $B == 0$

Definition at line 1189 of file `bignum.c`.

References `mpi_add_mpi()`, `MPI_CHK`, `mpi_cmp_int()`, `mpi_cmp_mpi()`, `mpi_div_mpi()`, and `mpi_sub_mpi()`.

Referenced by `mpi_exp_mod()`, `mpi_inv_mod()`, `mpi_is_prime()`, `rsa_check_privkey()`, `rsa_gen_key()`, and `rsa_private()`.

**13.95.4.22 int mpi\_msb (mpi \* X)**

Return the number of most significant bits.

Definition at line 200 of file `bignum.c`.

References `biL`, `mpi::n`, and `mpi::p`.

Referenced by `d2i_RSA_PUBKEY()`, `main()`, `mpi_div_mpi()`, `mpi_exp_mod()`, `mpi_gen_prime()`, `mpi_is_prime()`, `mpi_shift_l()`, `mpi_size()`, `mpi_write_string()`, `rsa_check_pubkey()`, `rsa_decryption()`, `rsa_encryption()`, and `rsa_gen_key()`.

**13.95.4.23 int mpi\_mul\_int (mpi \* X, mpi \* A, t\_int b)**

Baseline multiplication:  $X = A * b$ .

**Returns:**

0 if successful, 1 if memory allocation failed

Definition at line 992 of file `bignum.c`.

References `mpi_mul_mpi()`, `mpi::n`, `mpi::p`, and `mpi::s`.

Referenced by `mpi_div_mpi()`, and `mpi_read_string()`.

**13.95.4.24 int mpi\_mul\_mpi (mpi \* X, mpi \* A, mpi \* B)**

Baseline multiplication:  $X = A * B$ .

**Returns:**

0 if successful, 1 if memory allocation failed

Definition at line 956 of file `bignum.c`.

References `MPI_CHK`, `mpi_copy()`, `mpi_free()`, `mpi_grow()`, `mpi_init()`, `mpi_lset()`, `mpi_mul_hlp()`, `mpi::n`, `mpi::p`, and `mpi::s`.

Referenced by `main()`, `mpi_gcd()`, `mpi_is_prime()`, `mpi_mul_int()`, `mpi_self_test()`, `rsa_check_privkey()`, `rsa_gen_key()`, and `rsa_private()`.

**13.95.4.25 int mpi\_read\_binary (mpi \* X, unsigned char \* buf, int buflen)**

Import X from unsigned binary data, big endian.



**Parameters:**

*X* destination [mpi](#)  
*buf* input buffer  
*buflen* input buffer size

**Returns:**

0 if successful, 1 if memory allocation failed

Definition at line 484 of file bignum.c.

References CHARS\_TO\_LIMBS, ciL, MPI\_CHK, mpi\_grow(), mpi\_lset(), and mpi::p.

Referenced by asn1\_get\_mpi(), d2i\_RSA\_PUBKEY(), dhm\_read\_bignum(), dhm\_read\_public(), rsa\_private(), and rsa\_public().

**13.95.4.26 int mpi\_read\_file (mpi \* X, int radix, FILE \* fin)**

Read X from an opened file.

**Parameters:**

*X* destination [mpi](#)  
*radix* input numeric base  
*fin* input file handle

**Returns:**

0 if successful, or an XYSSL\_ERR\_MPI\_XXX error code

Definition at line 391 of file bignum.c.

References mpi\_get\_digit(), mpi\_read\_string(), and XYSSL\_ERR\_MPI\_FILE\_IO\_ERROR.

Referenced by main().

**13.95.4.27 int mpi\_read\_mystring (mpi \* X, int radix, char \* s)**

Definition at line 418 of file bignum.c.

References mpi\_get\_digit(), and mpi\_read\_string().

Referenced by rsa\_decryption(), and rsa\_encryption().

**13.95.4.28 int mpi\_read\_string (mpi \* X, int radix, char \* s)**

Import from an ASCII string.

**Parameters:**

*X* destination [mpi](#)  
*radix* input numeric base  
*s* null-terminated string buffer

**Returns:**

0 if successful, or an XYSSL\_ERR\_MPI\_XXX error code

Definition at line 243 of file bignum.c.

References BITS\_TO\_LIMBS, ciL, int, mpi\_add\_int(), MPI\_CHK, mpi\_free(), mpi\_get\_digit(), mpi\_grow(), mpi\_init(), mpi\_lset(), mpi\_mul\_int(), mpi::p, mpi::s, and XYSSL\_ERR\_MPI\_BAD\_INPUT\_DATA.

Referenced by main(), mpi\_read\_file(), mpi\_read\_mystring(), mpi\_self\_test(), rsa\_self\_test(), and ssl\_set\_dh\_param().

**13.95.4.29 int mpi\_self\_test (int verbose)**

Checkup routine.

**Returns:**

0 if successful, or 1 if the test failed

Definition at line 1854 of file bignum.c.

References MPI\_CHK, mpi\_cmp\_mpi(), mpi\_div\_mpi(), mpi\_exp\_mod(), mpi\_free(), mpi\_init(), mpi\_inv\_mod(), mpi\_mul\_mpi(), and mpi\_read\_string().

Referenced by main().

**13.95.4.30 int mpi\_shift\_l (mpi \* X, int count)**

Left-shift:  $X \ll= \text{count}$ .

**Returns:**

0 if successful, 1 if memory allocation failed

Definition at line 526 of file bignum.c.

References biL, BITS\_TO\_LIMBS, MPI\_CHK, mpi\_grow(), mpi\_msb(), mpi::n, and mpi::p.

Referenced by mpi\_div\_mpi(), mpi\_exp\_mod(), and mpi\_gen\_prime().

**13.95.4.31 int mpi\_shift\_r (mpi \* X, int count)**

Right-shift:  $X \gg= \text{count}$ .

**Returns:**

0 if successful, 1 if memory allocation failed

Definition at line 575 of file bignum.c.

References biL, mpi::n, and mpi::p.

Referenced by dhm\_make\_params(), dhm\_make\_public(), mpi\_div\_mpi(), mpi\_gcd(), mpi\_gen\_prime(), mpi\_inv\_mod(), and mpi\_is\_prime().

**13.95.4.32 int mpi\_size (mpi \* X)**

Return the total size in bytes.

Definition at line 218 of file bignum.c.

References mpi\_msb().

Referenced by dhm\_calc\_secret(), dhm\_make\_params(), dhm\_read\_params(), mpi\_write\_binary(), x509parse\_crt(), and x509parse\_key().

**13.95.4.33 int mpi\_sub\_abs (mpi \* X, mpi \* A, mpi \* B)**

Unsigned subtraction:  $X = |A| - |B|$ .

**Returns:**

0 if successful, XYSSL\_ERR\_MPI\_NEGATIVE\_VALUE if B is greater than A

Definition at line 761 of file bignum.c.

References MPI\_CHK, mpi\_cmp\_abs(), mpi\_copy(), mpi\_free(), mpi\_init(), mpi\_sub\_hlp(), mpi::n, mpi::p, and XYSSL\_ERR\_MPI\_NEGATIVE\_VALUE.

Referenced by mpi\_add\_mpi(), mpi\_gcd(), and mpi\_sub\_mpi().

**13.95.4.34 int mpi\_sub\_int (mpi \* X, mpi \* A, int b)**

Signed subtraction:  $X = A - b$ .

**Returns:**

0 if successful, 1 if memory allocation failed

Definition at line 876 of file bignum.c.

References mpi\_sub\_mpi(), mpi::n, mpi::p, and mpi::s.

Referenced by main(), mpi\_gen\_prime(), mpi\_is\_prime(), rsa\_check\_privkey(), and rsa\_gen\_key().

**13.95.4.35 int mpi\_sub\_mpi (mpi \* X, mpi \* A, mpi \* B)**

Signed subtraction:  $X = A - B$ .

**Returns:**

0 if successful, 1 if memory allocation failed

Definition at line 829 of file bignum.c.

References mpi\_add\_abs(), MPI\_CHK, mpi\_cmp\_abs(), mpi\_sub\_abs(), and mpi::s.

Referenced by mpi\_div\_mpi(), mpi\_inv\_mod(), mpi\_mod\_mpi(), mpi\_sub\_int(), and rsa\_private().

**13.95.4.36 void mpi\_swap (mpi \* X, mpi \* Y)**

Swap the contents of X and Y.

Definition at line 155 of file bignum.c.

Referenced by `rsa_gen_key()`.

### 13.95.4.37 `int mpi_write_binary (mpi * X, unsigned char * buf, int buflen)`

Export X into unsigned binary data, big endian.

#### Parameters:

*X* source [mpi](#)

*buf* output buffer

*buflen* output buffer size

#### Returns:

0 if successful, `XYSSL_ERR_MPI_BUFFER_TOO_SMALL` if buf isn't large enough

#### Note:

Call this function with `*buflen = 0` to obtain the minimum required buffer size in `*buflen`.

Definition at line 506 of file bignum.c.

References `ciL`, `mpi_size()`, `mpi::p`, and `XYSSL_ERR_MPI_BUFFER_TOO_SMALL`.

Referenced by `dhm_calc_secret()`, `dhm_make_public()`, `rsa_private()`, and `rsa_public()`.

### 13.95.4.38 `int mpi_write_file (char * p, mpi * X, int radix, FILE * fout)`

Write X into an opened file, or stdout.

#### Parameters:

*p* prefix, can be NULL

*X* source [mpi](#)

*radix* output numeric base

*fout* output file handle

#### Returns:

0 if successful, or an `XYSSL_ERR_MPI_XXX` error code

#### Note:

Set `fout == NULL` to print X on the console.

Definition at line 447 of file bignum.c.

References `MPI_CHK`, `mpi_write_string()`, and `XYSSL_ERR_MPI_FILE_IO_ERROR`.

Referenced by `generate_RSA_keys_ciphertext()`, `generate_RSA_keys_plaintext()`, and `main()`.

13.95

/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/bignum.h

File Reference

665

13.95.4.39 `int mpi_write_string (mpi * X, int radix, char * s, int * slen)`

---

Export into an ASCII string.

**Parameters:**

*X* source [mpi](#)

*radix* output numeric base

*s* string buffer

*slen* string buffer size

**Returns:**

0 if successful, or an XYSSL\_ERR\_MPI\_XXX error code

**Note:**

Call this function with *\*slen* = 0 to obtain the minimum required buffer size in *\*slen*.

Definition at line 328 of file bignum.c.

References [ciL](#), [MPI\\_CHK](#), [mpi\\_copy\(\)](#), [mpi\\_free\(\)](#), [mpi\\_init\(\)](#), [mpi\\_msb\(\)](#), [mpi\\_write\\_hlp\(\)](#), [mpi::n](#), [mpi::p](#), [mpi::s](#), [XYSSL\\_ERR\\_MPI\\_BAD\\_INPUT\\_DATA](#), and [XYSSL\\_ERR\\_MPI\\_BUFFER\\_TOO\\_SMALL](#).

Referenced by [mpi\\_write\\_file\(\)](#).

## 13.96 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/bn\_mul.h File Reference

```
#include "xyssl/config.h"
```

### Defines

- #define [MULADDC\\_INIT](#)
- #define [MULADDC\\_CORE](#)
- #define [MULADDC\\_STOP](#) }

### 13.96.1 Detailed Description

Definition in file [bn\\_mul.h](#).

### 13.96.2 Define Documentation

#### 13.96.2.1 #define MULADDC\_CORE

##### Value:

```
s0 = ( *s << biH ) >> biH;          \
s1 = ( *s >> biH ); s++;             \
rx = s0 * b1; r0 = s0 * b0;          \
ry = s1 * b0; r1 = s1 * b1;          \
r1 += ( rx >> biH );                  \
r1 += ( ry >> biH );                  \
rx <=< biH; ry <=< biH;                \
r0 += rx; r1 += (r0 < rx);            \
r0 += ry; r1 += (r0 < ry);            \
r0 += c; r1 += (r0 < c);              \
r0 += *d; r1 += (r0 < *d);            \
c = r1; *(d++) = r0;
```

Definition at line 662 of file [bn\\_mul.h](#).

Referenced by [mpi\\_mul\\_hlp\(\)](#).

#### 13.96.2.2 #define MULADDC\_INIT

##### Value:

```
{                                     \
    t_int s0, s1, b0, b1;           \
    t_int r0, r1, rx, ry;           \
    b0 = ( b << biH ) >> biH;       \
    b1 = ( b >> biH );
```

Definition at line 655 of file [bn\\_mul.h](#).

Referenced by [mpi\\_mul\\_hlp\(\)](#).

**13.96**

**/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/bn\_mul.h**

**File Reference**

**667**

**13.96.2.3 #define MULADDC\_STOP }**

---

Definition at line 676 of file bn\_mul.h.

Referenced by mpi\_mul\_hlp().

## 13.97 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/certs.h File Reference

### Variables

- char [test\\_ca\\_cert](#) []
- char [test\\_ca\\_key](#) []
- char [test\\_ca\\_pwd](#) []
- char [test\\_srv\\_cert](#) []
- char [test\\_srv\\_key](#) []
- char [test\\_cli\\_cert](#) []
- char [test\\_cli\\_key](#) []
- char [xyssl\\_ca\\_cert](#) []

### 13.97.1 Detailed Description

Definition in file [certs.h](#).

### 13.97.2 Variable Documentation

#### 13.97.2.1 char test\_ca\_cert[]

Definition at line 25 of file [certs.c](#).

Referenced by [main\(\)](#), [ssl\\_test\(\)](#), and [x509\\_self\\_test\(\)](#).

#### 13.97.2.2 char test\_ca\_key[]

Definition at line 49 of file [certs.c](#).

Referenced by [x509\\_self\\_test\(\)](#).

#### 13.97.2.3 char test\_ca\_pwd[]

Definition at line 81 of file [certs.c](#).

Referenced by [x509\\_self\\_test\(\)](#).

#### 13.97.2.4 char test\_cli\_cert[]

Definition at line 134 of file [certs.c](#).

Referenced by [main\(\)](#), and [x509\\_self\\_test\(\)](#).

#### 13.97.2.5 char test\_cli\_key[]

Definition at line 156 of file [certs.c](#).

Referenced by [main\(\)](#).



**13.97.2.6 char test\_srv\_cert[ ]**

Definition at line 83 of file certs.c.

Referenced by main(), and ssl\_test().

**13.97.2.7 char test\_srv\_key[ ]**

Definition at line 105 of file certs.c.

Referenced by main(), and ssl\_test().

**13.97.2.8 char xyssl\_ca\_cert[ ]**

Definition at line 185 of file certs.c.

Referenced by main().

## 13.98 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/debug.h File Reference

```
#include "xyssl/config.h"
#include "xyssl/ssl.h"
```

### Defines

- #define [SSL\\_DEBUG\\_MSG](#)(level, args) debug\_print\_msg( ssl, level, \_\_FILE\_\_, \_\_LINE\_\_, debug\_fmt args );
- #define [SSL\\_DEBUG\\_RET](#)(level, text, ret) debug\_print\_ret( ssl, level, \_\_FILE\_\_, \_\_LINE\_\_, text, ret );
- #define [SSL\\_DEBUG\\_BUF](#)(level, text, buf, len) debug\_print\_buf( ssl, level, \_\_FILE\_\_, \_\_LINE\_\_, text, buf, len );
- #define [SSL\\_DEBUG\\_MPI](#)(level, text, X) debug\_print\_mpi( ssl, level, \_\_FILE\_\_, \_\_LINE\_\_, text, X );
- #define [SSL\\_DEBUG\\_CRT](#)(level, text, crt) debug\_print\_crt( ssl, level, \_\_FILE\_\_, \_\_LINE\_\_, text, crt );

### Functions

- char \* [debug\\_fmt](#) (const char \*format,...)
- void [debug\\_print\\_msg](#) (ssl\_context \*ssl, int level, char \*file, int line, char \*text)
- void [debug\\_print\\_ret](#) (ssl\_context \*ssl, int level, char \*file, int line, char \*text, int ret)
- void [debug\\_print\\_buf](#) (ssl\_context \*ssl, int level, char \*file, int line, char \*text, unsigned char \*buf, int len)
- void [debug\\_print\\_mpi](#) (ssl\_context \*ssl, int level, char \*file, int line, char \*text, mpi \*X)
- void [debug\\_print\\_crt](#) (ssl\_context \*ssl, int level, char \*file, int line, char \*text, x509\_cert \*crt)

### 13.98.1 Detailed Description

Definition in file [debug.h](#).

### 13.98.2 Define Documentation

#### 13.98.2.1 #define [SSL\\_DEBUG\\_BUF](#)(level, text, buf, len) debug\_print\_buf( ssl, level, \_\_FILE\_\_, \_\_LINE\_\_, text, buf, len );

Definition at line 18 of file [debug.h](#).

Referenced by [ssl\\_calc\\_finished\(\)](#), [ssl\\_calc\\_verify\(\)](#), [ssl\\_decrypt\\_buf\(\)](#), [ssl\\_derive\\_keys\(\)](#), [ssl\\_encrypt\\_buf\(\)](#), [ssl\\_parse\\_client\\_hello\(\)](#), [ssl\\_parse\\_server\\_hello\(\)](#), [ssl\\_parse\\_server\\_key\\_exchange\(\)](#), [ssl\\_read\\_record\(\)](#), [ssl\\_write\\_certificate\\_request\(\)](#), [ssl\\_write\\_client\\_hello\(\)](#), [ssl\\_write\\_record\(\)](#), [ssl\\_write\\_server\\_hello\(\)](#), and [ssl\\_write\\_server\\_key\\_exchange\(\)](#).

```
13.98.2.2 #define SSL_DEBUG_CRT(level, text, crt) debug_print_crt( ssl, level, __FILE__,
    __LINE__, text, crt );
```

Definition at line 24 of file debug.h.

Referenced by ssl\_parse\_certificate(), and ssl\_write\_certificate().

```
13.98.2.3 #define SSL_DEBUG_MPI(level, text, X) debug_print_mpi( ssl, level, __FILE__,
    __LINE__, text, X );
```

Definition at line 21 of file debug.h.

Referenced by ssl\_parse\_client\_key\_exchange(), ssl\_parse\_server\_key\_exchange(), ssl\_write\_client\_key\_exchange(), and ssl\_write\_server\_key\_exchange().

```
13.98.2.4 #define SSL_DEBUG_MSG(level, args) debug_print_msg( ssl, level, __FILE__,
    __LINE__, debug_fmt args );
```

Definition at line 12 of file debug.h.

Referenced by ssl\_calc\_finished(), ssl\_calc\_verify(), ssl\_close\_notify(), ssl\_decrypt\_buf(), ssl\_derive\_keys(), ssl\_encrypt\_buf(), ssl\_fetch\_input(), ssl\_flush\_output(), ssl\_free(), ssl\_handshake(), ssl\_handshake\_client(), ssl\_handshake\_server(), ssl\_init(), ssl\_parse\_certificate(), ssl\_parse\_certificate\_request(), ssl\_parse\_certificate\_verify(), ssl\_parse\_change\_cipher\_spec(), ssl\_parse\_client\_hello(), ssl\_parse\_client\_key\_exchange(), ssl\_parse\_finished(), ssl\_parse\_server\_hello(), ssl\_parse\_server\_hello\_done(), ssl\_parse\_server\_key\_exchange(), ssl\_read(), ssl\_read\_record(), ssl\_write(), ssl\_write\_certificate(), ssl\_write\_certificate\_request(), ssl\_write\_certificate\_verify(), ssl\_write\_change\_cipher\_spec(), ssl\_write\_client\_hello(), ssl\_write\_client\_key\_exchange(), ssl\_write\_finished(), ssl\_write\_record(), ssl\_write\_server\_hello(), ssl\_write\_server\_hello\_done(), and ssl\_write\_server\_key\_exchange().

```
13.98.2.5 #define SSL_DEBUG_RET(level, text, ret) debug_print_ret( ssl, level, __FILE__,
    __LINE__, text, ret );
```

Definition at line 15 of file debug.h.

Referenced by ssl\_close\_notify(), ssl\_fetch\_input(), ssl\_flush\_output(), ssl\_parse\_certificate(), ssl\_parse\_certificate\_request(), ssl\_parse\_certificate\_verify(), ssl\_parse\_change\_cipher\_spec(), ssl\_parse\_client\_hello(), ssl\_parse\_client\_key\_exchange(), ssl\_parse\_finished(), ssl\_parse\_server\_hello(), ssl\_parse\_server\_hello\_done(), ssl\_parse\_server\_key\_exchange(), ssl\_read(), ssl\_read\_record(), ssl\_set\_dh\_param(), ssl\_write(), ssl\_write\_certificate(), ssl\_write\_certificate\_verify(), ssl\_write\_change\_cipher\_spec(), ssl\_write\_client\_hello(), ssl\_write\_client\_key\_exchange(), ssl\_write\_finished(), ssl\_write\_record(), ssl\_write\_server\_hello\_done(), and ssl\_write\_server\_key\_exchange().

## 13.98.3 Function Documentation

### 13.98.3.1 char\* debug\_fmt (const char \* *format*, ...)

Definition at line 38 of file debug.c.

**13.98.3.2 void debug\_print\_buf (ssl\_context \* ssl, int level, char \* file, int line, char \* text, unsigned char \* buf, int len)**

Definition at line 82 of file debug.c.

References `_ssl_context::f_dbg`, and `_ssl_context::p_dbg`.

**13.98.3.3 void debug\_print\_cert (ssl\_context \* ssl, int level, char \* file, int line, char \* text, x509\_cert \* crt)**

Definition at line 169 of file debug.c.

References `debug_print_mpi()`, `rsa_context::E`, `_ssl_context::f_dbg`, `rsa_context::N`, `_x509_cert::next`, `_ssl_context::p_dbg`, `_x509_cert::rsa`, and `x509parse_cert_info()`.

**13.98.3.4 void debug\_print\_mpi (ssl\_context \* ssl, int level, char \* file, int line, char \* text, mpi \* X)**

Definition at line 124 of file debug.c.

References `_ssl_context::f_dbg`, `mpi::n`, `mpi::p`, and `_ssl_context::p_dbg`.

Referenced by `debug_print_cert()`.

**13.98.3.5 void debug\_print\_msg (ssl\_context \* ssl, int level, char \* file, int line, char \* text)**

Definition at line 52 of file debug.c.

References `_ssl_context::f_dbg`, and `_ssl_context::p_dbg`.

**13.98.3.6 void debug\_print\_ret (ssl\_context \* ssl, int level, char \* file, int line, char \* text, int ret)**

Definition at line 66 of file debug.c.

References `_ssl_context::f_dbg`, and `_ssl_context::p_dbg`.

## 13.99 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/des.h File Reference

### Data Structures

- struct [des\\_context](#)  
*DES context structure.*
- struct [des3\\_context](#)  
*Triple-DES context structure.*

### Defines

- #define [DES\\_ENCRYPT](#) 1
- #define [DES\\_DECRYPT](#) 0

### Functions

- void [des\\_setkey\\_enc](#) ([des\\_context](#) \*ctx, unsigned char key[8])  
*DES key schedule (56-bit, encryption).*
- void [des\\_setkey\\_dec](#) ([des\\_context](#) \*ctx, unsigned char key[8])  
*DES key schedule (56-bit, decryption).*
- void [des3\\_set2key\\_enc](#) ([des3\\_context](#) \*ctx, unsigned char key[16])  
*Triple-DES key schedule (112-bit, encryption).*
- void [des3\\_set2key\\_dec](#) ([des3\\_context](#) \*ctx, unsigned char key[16])  
*Triple-DES key schedule (112-bit, decryption).*
- void [des3\\_set3key\\_enc](#) ([des3\\_context](#) \*ctx, unsigned char key[24])  
*Triple-DES key schedule (168-bit, encryption).*
- void [des3\\_set3key\\_dec](#) ([des3\\_context](#) \*ctx, unsigned char key[24])  
*Triple-DES key schedule (168-bit, decryption).*
- void [des\\_crypt\\_ecb](#) ([des\\_context](#) \*ctx, unsigned char input[8], unsigned char output[8])  
*DES-ECB block encryption/decryption.*
- void [des\\_crypt\\_cbc](#) ([des\\_context](#) \*ctx, int mode, int length, unsigned char iv[8], unsigned char \*input, unsigned char \*output)  
*DES-CBC buffer encryption/decryption.*
- void [des3\\_crypt\\_ecb](#) ([des3\\_context](#) \*ctx, unsigned char input[8], unsigned char output[8])  
*3DES-ECB block encryption/decryption*

- void `des3_crypt_cbc` (`des3_context` \*ctx, int mode, int length, unsigned char iv[8], unsigned char \*input, unsigned char \*output)  
*3DES-CBC buffer encryption/decryption*
- int `des_self_test` (int verbose)

### 13.99.1 Detailed Description

Definition in file [des.h](#).

### 13.99.2 Define Documentation

#### 13.99.2.1 #define DES\_DECRYPT 0

Definition at line 8 of file [des.h](#).

Referenced by `des_self_test()`, `ssl_decrypt_buf()`, and `x509_des3_decrypt()`.

#### 13.99.2.2 #define DES\_ENCRYPT 1

Definition at line 7 of file [des.h](#).

Referenced by `des3_crypt_cbc()`, `des_crypt_cbc()`, `main()`, and `ssl_encrypt_buf()`.

### 13.99.3 Function Documentation

#### 13.99.3.1 void `des3_crypt_cbc` (`des3_context` \*ctx, int mode, int length, unsigned char iv[8], unsigned char \*input, unsigned char \*output)

3DES-CBC buffer encryption/decryption

##### Parameters:

*ctx* 3DES context  
*mode* DES\_ENCRYPT or DES\_DECRYPT  
*length* length of the input data  
*iv* initialization vector (updated after use)  
*input* buffer holding the input data  
*output* buffer holding the output data

Definition at line 593 of file [des.c](#).

References `des3_crypt_ecb()`, and `DES_ENCRYPT`.

Referenced by `des_self_test()`, `main()`, `ssl_decrypt_buf()`, `ssl_encrypt_buf()`, and `x509_des3_decrypt()`.

#### 13.99.3.2 void `des3_crypt_ecb` (`des3_context` \*ctx, unsigned char input[8], unsigned char output[8])

3DES-ECB block encryption/decryption

**Parameters:**

*ctx* 3DES context  
*input* 64-bit input block  
*output* 64-bit output block

Definition at line 552 of file des.c.

References DES\_FP, DES\_IP, DES\_ROUND, GET\_ULONG\_BE, PUT\_ULONG\_BE, and des3\_context::sk.

Referenced by des3\_crypt\_cbc(), and des\_self\_test().

**13.99.3.3 void des3\_set2key\_dec (des3\_context \* ctx, unsigned char key[16])**

Triple-DES key schedule (112-bit, decryption).

**Parameters:**

*ctx* 3DES context to be initialized  
*key* 16-byte secret key

Definition at line 420 of file des.c.

References des3\_set2key(), and des3\_context::sk.

Referenced by des\_self\_test().

**13.99.3.4 void des3\_set2key\_enc (des3\_context \* ctx, unsigned char key[16])**

Triple-DES key schedule (112-bit, encryption).

**Parameters:**

*ctx* 3DES context to be initialized  
*key* 16-byte secret key

Definition at line 409 of file des.c.

References des3\_set2key(), and des3\_context::sk.

Referenced by des\_self\_test().

**13.99.3.5 void des3\_set3key\_dec (des3\_context \* ctx, unsigned char key[24])**

Triple-DES key schedule (168-bit, decryption).

**Parameters:**

*ctx* 3DES context to be initialized  
*key* 24-byte secret key

Definition at line 465 of file des.c.

References des3\_set3key(), and des3\_context::sk.

Referenced by des\_self\_test(), ssl\_derive\_keys(), and x509\_des3\_decrypt().

**13.99.3.6 void des3\_set3key\_enc (des3\_context \* ctx, unsigned char key[24])**

Triple-DES key schedule (168-bit, encryption).

**Parameters:**

*ctx* 3DES context to be initialized

*key* 24-byte secret key

Definition at line 454 of file des.c.

References des3\_set3key(), and des3\_context::sk.

Referenced by des\_self\_test(), main(), and ssl\_derive\_keys().

**13.99.3.7 void des\_crypt\_cbc (des\_context \* ctx, int mode, int length, unsigned char iv[8], unsigned char \* input, unsigned char \* output)**

DES-CBC buffer encryption/decryption.

**Parameters:**

*ctx* DES context

*mode* DES\_ENCRYPT or DES\_DECRYPT

*length* length of the input data

*iv* initialization vector (updated after use)

*input* buffer holding the input data

*output* buffer holding the output data

Definition at line 505 of file des.c.

References des\_crypt\_ecb(), and DES\_ENCRYPT.

Referenced by des\_self\_test(), and main().

**13.99.3.8 void des\_crypt\_ecb (des\_context \* ctx, unsigned char input[8], unsigned char output[8])**

DES-ECB block encryption/decryption.

**Parameters:**

*ctx* DES context

*input* 64-bit input block

*output* 64-bit output block

Definition at line 476 of file des.c.

References DES\_FP, DES\_IP, DES\_ROUND, GET\_ULONG\_BE, PUT\_ULONG\_BE, and des\_context::sk.

Referenced by des\_crypt\_cbc(), and des\_self\_test().



### 13.99.3.9 int des\_self\_test (int *verbose*)

Definition at line 694 of file des.c.

References `buf`, `des3_crypt_cbc()`, `des3_crypt_ecb()`, `des3_set2key_dec()`, `des3_set2key_enc()`, `des3_set3key_dec()`, `des3_set3key_enc()`, `des3_test_buf`, `des3_test_cbc_dec`, `des3_test_cbc_enc`, `des3_test_ecb_dec`, `des3_test_ecb_enc`, `des3_test_iv`, `des3_test_keys`, `des_crypt_cbc()`, `des_crypt_ecb()`, `DES_DECRYPT`, `des_setkey_dec()`, `des_setkey_enc()`, and `prv`.

Referenced by `main()`.

### 13.99.3.10 void des\_setkey\_dec (des\_context \* *ctx*, unsigned char *key*[8])

DES key schedule (56-bit, decryption).

#### Parameters:

*ctx* DES context to be initialized

*key* 8-byte secret key

Definition at line 368 of file des.c.

References `des_setkey()`, `des_context::sk`, and `SWAP`.

Referenced by `des_self_test()`.

### 13.99.3.11 void des\_setkey\_enc (des\_context \* *ctx*, unsigned char *key*[8])

DES key schedule (56-bit, encryption).

#### Parameters:

*ctx* DES context to be initialized

*key* 8-byte secret key

Definition at line 360 of file des.c.

References `des_setkey()`, and `des_context::sk`.

Referenced by `des_self_test()`, and `main()`.

## 13.100 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/dhm.h File Reference

```
#include "bignum.h"
```

### Data Structures

- struct [dhm\\_context](#)

### Defines

- #define [XYSSL\\_ERR\\_DHM\\_BAD\\_INPUT\\_DATA](#) -0x0480
- #define [XYSSL\\_ERR\\_DHM\\_READ\\_PARAMS\\_FAILED](#) -0x0490
- #define [XYSSL\\_ERR\\_DHM\\_MAKE\\_PARAMS\\_FAILED](#) -0x04A0
- #define [XYSSL\\_ERR\\_DHM\\_READ\\_PUBLIC\\_FAILED](#) -0x04B0
- #define [XYSSL\\_ERR\\_DHM\\_MAKE\\_PUBLIC\\_FAILED](#) -0x04C0
- #define [XYSSL\\_ERR\\_DHM\\_CALC\\_SECRET\\_FAILED](#) -0x04D0

### Functions

- [int dhm\\_read\\_params](#) ([dhm\\_context](#) \*ctx, unsigned char \*\*p, unsigned char \*end)  
*Parse the ServerKeyExchange parameters.*
- [int dhm\\_make\\_params](#) ([dhm\\_context](#) \*ctx, [int](#) s\_size, unsigned char \*output, [int](#) \*olen, [int](#)(\*f\_rng)(void \*), void \*p\_rng)  
*Setup and write the ServerKeyExchange parameters.*
- [int dhm\\_read\\_public](#) ([dhm\\_context](#) \*ctx, unsigned char \*input, [int](#) ilen)  
*Import the peer's public value  $G^Y$ .*
- [int dhm\\_make\\_public](#) ([dhm\\_context](#) \*ctx, [int](#) s\_size, unsigned char \*output, [int](#) olen, [int](#)(\*f\_rng)(void \*), void \*p\_rng)  
*Create own private value  $X$  and export  $G^X$ .*
- [int dhm\\_calc\\_secret](#) ([dhm\\_context](#) \*ctx, unsigned char \*output, [int](#) \*olen)  
*Derive and export the shared secret  $(G^Y)^X \bmod P$ .*
- void [dhm\\_free](#) ([dhm\\_context](#) \*ctx)
- [int dhm\\_self\\_test](#) ([int](#) verbose)  
*Checkup routine.*

### 13.100.1 Detailed Description

Definition in file [dhm.h](#).

**13.100.2 Define Documentation**

---

**13.100.2.1 #define XYSSL\_ERR\_DHM\_BAD\_INPUT\_DATA -0x0480**

Definition at line 9 of file dhm.h.

Referenced by dhm\_calc\_secret(), dhm\_make\_public(), dhm\_read\_bignum(), dhm\_read\_params(), and dhm\_read\_public().

**13.100.2.2 #define XYSSL\_ERR\_DHM\_CALC\_SECRET\_FAILED -0x04D0**

Definition at line 14 of file dhm.h.

Referenced by dhm\_calc\_secret().

**13.100.2.3 #define XYSSL\_ERR\_DHM\_MAKE\_PARAMS\_FAILED -0x04A0**

Definition at line 11 of file dhm.h.

Referenced by dhm\_make\_params().

**13.100.2.4 #define XYSSL\_ERR\_DHM\_MAKE\_PUBLIC\_FAILED -0x04C0**

Definition at line 13 of file dhm.h.

Referenced by dhm\_make\_public().

**13.100.2.5 #define XYSSL\_ERR\_DHM\_READ\_PARAMS\_FAILED -0x0490**

Definition at line 10 of file dhm.h.

Referenced by dhm\_read\_bignum().

**13.100.2.6 #define XYSSL\_ERR\_DHM\_READ\_PUBLIC\_FAILED -0x04B0**

Definition at line 12 of file dhm.h.

Referenced by dhm\_read\_public().

**13.100.3 Function Documentation****13.100.3.1 int dhm\_calc\_secret (dhm\_context \* *ctx*, unsigned char \* *output*, int \* *olen*)**

Derive and export the shared secret  $(G^Y)^X \bmod P$ .

**Parameters:**

*ctx* DHM context

*output* destination buffer

*olen* number of chars written

**Returns:**

0 if successful, or an XYSSL\_ERR\_DHM\_XXX error code

Definition at line 208 of file dhm.c.

References dhm\_context::GY, dhm\_context::K, MPI\_CHK, mpi\_exp\_mod(), mpi\_size(), mpi\_write\_binary(), dhm\_context::P, dhm\_context::RP, dhm\_context::X, XYSSL\_ERR\_DHM\_BAD\_INPUT\_DATA, and XYSSL\_ERR\_DHM\_CALC\_SECRET\_FAILED.

Referenced by main(), ssl\_parse\_client\_key\_exchange(), and ssl\_write\_client\_key\_exchange().

**13.100.3.2 void dhm\_free (dhm\_context \* ctx)**

Definition at line 234 of file dhm.c.

References dhm\_context::G, dhm\_context::GX, dhm\_context::GY, dhm\_context::K, mpi\_free(), dhm\_context::P, dhm\_context::RP, and dhm\_context::X.

Referenced by main(), and ssl\_free().

**13.100.3.3 int dhm\_make\_params (dhm\_context \* ctx, int s\_size, unsigned char \* output, int \* olen, int(\*) (void \*) f\_rng, void \* p\_rng)**

Setup and write the ServerKeyExchange parameters.

**Parameters:**

*ctx* DHM context  
*x\_size* private value size in bits  
*output* destination buffer  
*olen* number of chars written  
*f\_rng* RNG function  
*p\_rng* RNG parameter

**Note:**

This function assumes that ctx->P and ctx->G have already been properly set (for example using mpi\_read\_string or mpi\_read\_binary).

**Returns:**

0 if successful, or an XYSSL\_ERR\_DHM\_XXX error code

Definition at line 93 of file dhm.c.

References DHM\_MPI\_EXPORT, dhm\_context::G, dhm\_context::GX, dhm\_context::len, MPI\_CHK, mpi\_cmp\_mpi(), mpi\_exp\_mod(), mpi\_grow(), mpi\_lset(), mpi\_shift\_r(), mpi\_size(), dhm\_context::P, mpi::p, dhm\_context::RP, dhm\_context::X, and XYSSL\_ERR\_DHM\_MAKE\_PARAMS\_FAILED.

Referenced by main(), and ssl\_write\_server\_key\_exchange().

**13.100.3.4 int dhm\_make\_public (dhm\_context \* ctx, int s\_size, unsigned char \* output, int olen, int(\*) (void \*) f\_rng, void \* p\_rng)**

Create own private value X and export  $G^X$ .

### 13.100

/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/dhm.h File

#### Reference Parameters:

681

*ctx* DHM context  
*x\_size* private value size in bits  
*output* destination buffer  
*olen* must be equal to *ctx->P.len*  
*f\_rng* RNG function  
*p\_rng* RNG parameter

#### Returns:

0 if successful, or an XYSSL\_ERR\_DHM\_XXX error code

Definition at line 167 of file dhm.c.

References *dhm\_context::G*, *dhm\_context::GX*, *dhm\_context::len*, *MPI\_CHK*, *mpi\_cmp\_mpi()*, *mpi\_exp\_mod()*, *mpi\_grow()*, *mpi\_lset()*, *mpi\_shift\_r()*, *mpi\_write\_binary()*, *dhm\_context::P*, *mpi::p*, *dhm\_context::RP*, *dhm\_context::X*, *XYSSL\_ERR\_DHM\_BAD\_INPUT\_DATA*, and *XYSSL\_ERR\_DHM\_MAKE\_PUBLIC\_FAILED*.

Referenced by *main()*, and *ssl\_write\_client\_key\_exchange()*.

#### 13.100.3.5 int dhm\_read\_params (dhm\_context \* ctx, unsigned char \*\* p, unsigned char \* end)

Parse the ServerKeyExchange parameters.

#### Parameters:

*ctx* DHM context  
*p* &(start of input buffer)  
*end* end of buffer

#### Returns:

0 if successful, or an XYSSL\_ERR\_DHM\_XXX error code

Definition at line 63 of file dhm.c.

References *dhm\_read\_bignum()*, *dhm\_context::G*, *dhm\_context::GY*, *dhm\_context::len*, *mpi\_size()*, *dhm\_context::P*, and *XYSSL\_ERR\_DHM\_BAD\_INPUT\_DATA*.

Referenced by *main()*, and *ssl\_parse\_server\_key\_exchange()*.

#### 13.100.3.6 int dhm\_read\_public (dhm\_context \* ctx, unsigned char \* input, int ilen)

Import the peer's public value  $G^Y$ .

#### Parameters:

*ctx* DHM context  
*input* input buffer  
*ilen* size of buffer

#### Returns:

0 if successful, or an XYSSL\_ERR\_DHM\_XXX error code

Definition at line 150 of file dhm.c.

References dhm\_context::GY, dhm\_context::len, mpi\_read\_binary(), XYSSL\_ERR\_DHM\_BAD\_INPUT\_DATA, and XYSSL\_ERR\_DHM\_READ\_PUBLIC\_FAILED.

Referenced by main(), and ssl\_parse\_client\_key\_exchange().

### **13.100.3.7 int dhm\_self\_test (int *verbose*)**

Checkup routine.

#### **Returns:**

0 if successful, or 1 if the test failed

Definition at line 246 of file dhm.c.

## Data Structures

- struct [havege\\_state](#)  
*HAVEGE state structure.*

## Defines

- #define [COLLECT\\_SIZE](#) 1024

## Functions

- void [havege\\_init](#) ([havege\\_state](#) \*hs)  
*HAVEGE initialization.*
- int [havege\\_rand](#) (void \*p\_rng)  
*HAVEGE rand function.*

### 13.101.1 Detailed Description

Definition in file [havege.h](#).

### 13.101.2 Define Documentation

#### 13.101.2.1 #define COLLECT\_SIZE 1024

Definition at line 7 of file [havege.h](#).

Referenced by [havege\\_fill\(\)](#), and [havege\\_rand\(\)](#).

### 13.101.3 Function Documentation

#### 13.101.3.1 void havege\_init (havege\_state \* hs)

HAVEGE initialization.

##### Parameters:

*hs* HAVEGE state to be initialized

Definition at line 188 of file [havege.c](#).

References [havege\\_fill\(\)](#).

Referenced by [generate\\_AES\\_key\(\)](#), [generate\\_RSA\\_keys\\_ciphertext\(\)](#), [generate\\_RSA\\_keys\\_plaintext\(\)](#), [initiate\\_migration\\_process\(\)](#), [main\(\)](#), and [ssl\\_test\(\)](#).

**13.101.3.2 int havege\_rand (void \* *p\_rng*)**

HAVEGE rand function.

**Parameters:**

*rng\_st* points to an HAVEGE state

**Returns:**

A random int

Definition at line 198 of file havege.c.

References COLLECT\_SIZE, havege\_fill(), havege\_state::offset, and havege\_state::pool.

Referenced by generate\_AES\_key(), generate\_RSA\_keys\_ciphertext(), generate\_RSA\_keys\_plaintext(), initiate\_migration\_process(), main(), and ssl\_test().



## 13.102 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/md2.h File Reference

### Data Structures

- struct [md2\\_context](#)  
*MD2 context structure.*

### Functions

- void [md2\\_starts](#) ([md2\\_context](#) \*ctx)  
*MD2 context setup.*
- void [md2\\_update](#) ([md2\\_context](#) \*ctx, unsigned char \*input, [int](#) ilen)  
*MD2 process buffer.*
- void [md2\\_finish](#) ([md2\\_context](#) \*ctx, unsigned char output[16])  
*MD2 final digest.*
- void [md2](#) (unsigned char \*input, [int](#) ilen, unsigned char output[16])  
*Output = MD2( input buffer ).*
- [int](#) [md2\\_file](#) (char \*path, unsigned char output[16])  
*Output = MD2( file contents ).*
- void [md2\\_hmac\\_starts](#) ([md2\\_context](#) \*ctx, unsigned char \*key, [int](#) keylen)  
*MD2 HMAC context setup.*
- void [md2\\_hmac\\_update](#) ([md2\\_context](#) \*ctx, unsigned char \*input, [int](#) ilen)  
*MD2 HMAC process buffer.*
- void [md2\\_hmac\\_finish](#) ([md2\\_context](#) \*ctx, unsigned char output[16])  
*MD2 HMAC final digest.*
- void [md2\\_hmac](#) (unsigned char \*key, [int](#) keylen, unsigned char \*input, [int](#) ilen, unsigned char output[16])  
*Output = HMAC-MD2( hmac key, input buffer ).*
- [int](#) [md2\\_self\\_test](#) ([int](#) verbose)  
*Checkup routine.*

#### 13.102.1 Detailed Description

Definition in file [md2.h](#).

### 13.102.2 Function Documentation

#### 13.102.2.1 void md2 (unsigned char \* *input*, int *ilen*, unsigned char *output*[16])

Output = MD2( input buffer ).

**Parameters:**

*input* buffer holding the data  
*ilen* length of the input data  
*output* MD2 checksum result

Referenced by x509\_hash().

#### 13.102.2.2 int md2\_file (char \* *path*, unsigned char *output*[16])

Output = MD2( file contents ).

**Parameters:**

*path* input file name  
*output* MD2 checksum result

**Returns:**

0 if successful, 1 if fopen failed, or 2 if fread failed

#### 13.102.2.3 void md2\_finish (md2\_context \* *ctx*, unsigned char *output*[16])

MD2 final digest.

**Parameters:**

*ctx* MD2 context  
*output* MD2 checksum result

#### 13.102.2.4 void md2\_hmac (unsigned char \* *key*, int *keylen*, unsigned char \* *input*, int *ilen*, unsigned char *output*[16])

Output = HMAC-MD2( hmac key, input buffer ).

**Parameters:**

*key* HMAC secret key  
*keylen* length of the HMAC key  
*input* buffer holding the data  
*ilen* length of the input data  
*output* HMAC-MD2 result

## 13.102

/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/md2.h File

### Reference

687

**13.102.2.5** `void md2_hmac_finish (md2_context * ctx, unsigned char output[16])`

MD2 HMAC final digest.

#### Parameters:

*ctx* HMAC context

*output* MD2 HMAC checksum result

**13.102.2.6** `void md2_hmac_starts (md2_context * ctx, unsigned char * key, int keylen)`

MD2 HMAC context setup.

#### Parameters:

*ctx* HMAC context to be initialized

*key* HMAC secret key

*keylen* length of the HMAC key

**13.102.2.7** `void md2_hmac_update (md2_context * ctx, unsigned char * input, int ilen)`

MD2 HMAC process buffer.

#### Parameters:

*ctx* HMAC context

*input* buffer holding the data

*ilen* length of the input data

**13.102.2.8** `int md2_self_test (int verbose)`

Checkup routine.

#### Returns:

0 if successful, or 1 if the test failed

Referenced by main().

**13.102.2.9** `void md2_starts (md2_context * ctx)`

MD2 context setup.

#### Parameters:

*ctx* context to be initialized

**13.102.2.10** void md2\_update (md2\_context \* *ctx*, unsigned char \* *input*, int *ilen*)

MD2 process buffer.

**Parameters:**

*ctx* MD2 context

*input* buffer holding the data

*ilen* length of the input data

## 13.103 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/md4.h File Reference

### Data Structures

- struct [md4\\_context](#)  
*MD4 context structure.*

### Functions

- void [md4\\_starts](#) ([md4\\_context](#) \*ctx)  
*MD4 context setup.*
- void [md4\\_update](#) ([md4\\_context](#) \*ctx, unsigned char \*input, [int](#) ilen)  
*MD4 process buffer.*
- void [md4\\_finish](#) ([md4\\_context](#) \*ctx, unsigned char output[16])  
*MD4 final digest.*
- void [md4](#) (unsigned char \*input, [int](#) ilen, unsigned char output[16])  
*Output = MD4( input buffer ).*
- [int](#) [md4\\_file](#) (char \*path, unsigned char output[16])  
*Output = MD4( file contents ).*
- void [md4\\_hmac\\_starts](#) ([md4\\_context](#) \*ctx, unsigned char \*key, [int](#) keylen)  
*MD4 HMAC context setup.*
- void [md4\\_hmac\\_update](#) ([md4\\_context](#) \*ctx, unsigned char \*input, [int](#) ilen)  
*MD4 HMAC process buffer.*
- void [md4\\_hmac\\_finish](#) ([md4\\_context](#) \*ctx, unsigned char output[16])  
*MD4 HMAC final digest.*
- void [md4\\_hmac](#) (unsigned char \*key, [int](#) keylen, unsigned char \*input, [int](#) ilen, unsigned char output[16])  
*Output = HMAC-MD4( hmac key, input buffer ).*
- [int](#) [md4\\_self\\_test](#) ([int](#) verbose)  
*Checkup routine.*

### 13.103.1 Detailed Description

Definition in file [md4.h](#).

### 13.103.2 Function Documentation

#### 13.103.2.1 void md4 (unsigned char \* *input*, int *ilen*, unsigned char *output*[16])

Output = MD4( input buffer ).

**Parameters:**

*input* buffer holding the data  
*ilen* length of the input data  
*output* MD4 checksum result

Referenced by main(), and x509\_hash().

#### 13.103.2.2 int md4\_file (char \* *path*, unsigned char *output*[16])

Output = MD4( file contents ).

**Parameters:**

*path* input file name  
*output* MD4 checksum result

**Returns:**

0 if successful, 1 if fopen failed, or 2 if fread failed

#### 13.103.2.3 void md4\_finish (md4\_context \* *ctx*, unsigned char *output*[16])

MD4 final digest.

**Parameters:**

*ctx* MD4 context  
*output* MD4 checksum result

#### 13.103.2.4 void md4\_hmac (unsigned char \* *key*, int *keylen*, unsigned char \* *input*, int *ilen*, unsigned char *output*[16])

Output = HMAC-MD4( hmac key, input buffer ).

**Parameters:**

*key* HMAC secret key  
*keylen* length of the HMAC key  
*input* buffer holding the data  
*ilen* length of the input data  
*output* HMAC-MD4 result

---

**13.103.2.5 void md4\_hmac\_finish (md4\_context \* *ctx*, unsigned char *output*[16])**

MD4 HMAC final digest.

**Parameters:**

*ctx* HMAC context

*output* MD4 HMAC checksum result

**13.103.2.6 void md4\_hmac\_starts (md4\_context \* *ctx*, unsigned char \* *key*, int *keylen*)**

MD4 HMAC context setup.

**Parameters:**

*ctx* HMAC context to be initialized

*key* HMAC secret key

*keylen* length of the HMAC key

**13.103.2.7 void md4\_hmac\_update (md4\_context \* *ctx*, unsigned char \* *input*, int *ilen*)**

MD4 HMAC process buffer.

**Parameters:**

*ctx* HMAC context

*input* buffer holding the data

*ilen* length of the input data

**13.103.2.8 int md4\_self\_test (int *verbose*)**

Checkup routine.

**Returns:**

0 if successful, or 1 if the test failed

Referenced by main().

**13.103.2.9 void md4\_starts (md4\_context \* *ctx*)**

MD4 context setup.

**Parameters:**

*ctx* context to be initialized

**13.103.2.10** void md4\_update (md4\_context \* *ctx*, unsigned char \* *input*, int *ilen*)

MD4 process buffer.

**Parameters:**

*ctx* MD4 context

*input* buffer holding the data

*ilen* length of the input data



## 13.104 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/md5.h File Reference

### Data Structures

- struct [md5\\_context](#)  
*MD5 context structure.*

### Functions

- void [md5\\_starts](#) ([md5\\_context](#) \*ctx)  
*MD5 context setup.*
- void [md5\\_update](#) ([md5\\_context](#) \*ctx, unsigned char \*input, [int](#) ilen)  
*MD5 process buffer.*
- void [md5\\_finish](#) ([md5\\_context](#) \*ctx, unsigned char output[16])  
*MD5 final digest.*
- void [md5](#) (unsigned char \*input, [int](#) ilen, unsigned char output[16])  
*Output = MD5( input buffer ).*
- [int](#) [md5\\_file](#) (char \*path, unsigned char output[16])  
*Output = MD5( file contents ).*
- void [md5\\_hmac\\_starts](#) ([md5\\_context](#) \*ctx, unsigned char \*key, [int](#) keylen)  
*MD5 HMAC context setup.*
- void [md5\\_hmac\\_update](#) ([md5\\_context](#) \*ctx, unsigned char \*input, [int](#) ilen)  
*MD5 HMAC process buffer.*
- void [md5\\_hmac\\_finish](#) ([md5\\_context](#) \*ctx, unsigned char output[16])  
*MD5 HMAC final digest.*
- void [md5\\_hmac](#) (unsigned char \*key, [int](#) keylen, unsigned char \*input, [int](#) ilen, unsigned char output[16])  
*Output = HMAC-MD5( hmac key, input buffer ).*
- [int](#) [md5\\_self\\_test](#) ([int](#) verbose)  
*Checkup routine.*

### 13.104.1 Detailed Description

Definition in file [md5.h](#).

## 13.104.2 Function Documentation

### 13.104.2.1 void md5 (unsigned char \* *input*, int *ilen*, unsigned char *output*[16])

Output = MD5( *input* buffer ).

**Parameters:**

*input* buffer holding the data  
*ilen* length of the input data  
*output* MD5 checksum result

Definition at line 278 of file md5.c.

References md5\_finish(), md5\_starts(), and md5\_update().

Referenced by initiate\_migration\_process(), main(), md5\_hmac\_starts(), md5\_self\_test(), reply\_migration\_process(), ssl\_calc\_verify(), ssl\_derive\_keys(), ssl\_mac\_md5(), ssl\_parse\_finished(), ssl\_parse\_server\_key\_exchange(), ssl\_write\_finished(), ssl\_write\_server\_key\_exchange(), and x509\_hash().

### 13.104.2.2 int md5\_file (char \* *path*, unsigned char *output*[16])

Output = MD5( *file* contents ).

**Parameters:**

*path* input file name  
*output* MD5 checksum result

**Returns:**

0 if successful, 1 if fopen failed, or 2 if fread failed

Definition at line 292 of file md5.c.

References buf, f, md5\_finish(), md5\_starts(), and md5\_update().

Referenced by md5\_wrapper().

### 13.104.2.3 void md5\_finish (md5\_context \* *ctx*, unsigned char *output*[16])

MD5 final digest.

**Parameters:**

*ctx* MD5 context  
*output* MD5 checksum result

Definition at line 250 of file md5.c.

References md5\_padding, md5\_update(), PUT\_ULONG\_LE, md5\_context::state, and md5\_context::total.

Referenced by md5(), md5\_file(), md5\_hmac\_finish(), ssl\_calc\_finished(), ssl\_calc\_verify(), ssl\_derive\_keys(), ssl\_mac\_md5(), ssl\_parse\_server\_key\_exchange(), ssl\_write\_server\_key\_exchange(), and x509\_des3\_decrypt().

### 13.104

/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/md5.h File Reference 695

**13.104.2.4** `void md5_hmac (unsigned char * key, int keylen, unsigned char * input, int ilen, unsigned char output[16])`

Output = HMAC-MD5( hmac key, input buffer ).

#### Parameters:

*key* HMAC secret key

*keylen* length of the HMAC key

*input* buffer holding the data

*ilen* length of the input data

*output* HMAC-MD5 result

Definition at line 378 of file md5.c.

References md5\_hmac\_finish(), md5\_hmac\_starts(), and md5\_hmac\_update().

Referenced by ssl\_decrypt\_buf(), ssl\_encrypt\_buf(), and tls1\_prf().

**13.104.2.5** `void md5_hmac_finish (md5_context * ctx, unsigned char output[16])`

MD5 HMAC final digest.

#### Parameters:

*ctx* HMAC context

*output* MD5 HMAC checksum result

Definition at line 362 of file md5.c.

References md5\_finish(), md5\_starts(), md5\_update(), and md5\_context::opad.

Referenced by md5\_hmac(), and md5\_self\_test().

**13.104.2.6** `void md5_hmac_starts (md5_context * ctx, unsigned char * key, int keylen)`

MD5 HMAC context setup.

#### Parameters:

*ctx* HMAC context to be initialized

*key* HMAC secret key

*keylen* length of the HMAC key

Definition at line 324 of file md5.c.

References md5\_context::ipad, md5(), md5\_starts(), md5\_update(), and md5\_context::opad.

Referenced by md5\_hmac(), and md5\_self\_test().

**13.104.2.7** `void md5_hmac_update (md5_context * ctx, unsigned char * input, int ilen)`

MD5 HMAC process buffer.

**Parameters:**

*ctx* HMAC context

*input* buffer holding the data

*ilen* length of the input data

Definition at line 354 of file md5.c.

References md5\_update().

Referenced by md5\_hmac(), and md5\_self\_test().

**13.104.2.8 int md5\_self\_test (int verbose)**

Checksum routine.

**Returns:**

0 if successful, or 1 if the test failed

Definition at line 495 of file md5.c.

References buf, md5(), md5\_hmac\_finish(), md5\_hmac\_starts(), md5\_hmac\_test\_buf, md5\_hmac\_test\_buflen, md5\_hmac\_test\_key, md5\_hmac\_test\_keylen, md5\_hmac\_test\_sum, md5\_hmac\_update(), md5\_test\_buf, md5\_test\_buflen, and md5\_test\_sum.

Referenced by main().

**13.104.2.9 void md5\_starts (md5\_context \* ctx)**

MD5 context setup.

**Parameters:**

*ctx* context to be initialized

Definition at line 61 of file md5.c.

References md5\_context::state, and md5\_context::total.

Referenced by md5(), md5\_file(), md5\_hmac\_finish(), md5\_hmac\_starts(), ssl\_calc\_finished(), ssl\_calc\_verify(), ssl\_derive\_keys(), ssl\_init(), ssl\_mac\_md5(), ssl\_parse\_server\_key\_exchange(), ssl\_write\_server\_key\_exchange(), and x509\_des3\_decrypt().

**13.104.2.10 void md5\_update (md5\_context \* ctx, unsigned char \* input, int ilen)**

MD5 process buffer.

**Parameters:**

*ctx* MD5 context

*input* buffer holding the data

*ilen* length of the input data

Definition at line 198 of file md5.c.

### 13.104

**/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/md5.h File Reference**

**697**

---

References `md5_context::buffer`, `md5_process()`, and `md5_context::total`.

Referenced by `md5()`, `md5_file()`, `md5_finish()`, `md5_hmac_finish()`, `md5_hmac_starts()`, `md5_hmac_update()`, `ssl_calc_finished()`, `ssl_calc_verify()`, `ssl_derive_keys()`, `ssl_mac_md5()`, `ssl_parse_client_hello()`, `ssl_parse_server_key_exchange()`, `ssl_read_record()`, `ssl_write_record()`, `ssl_write_server_key_exchange()`, and `x509_des3_decrypt()`.

## 13.105 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/net.h File Reference

### Defines

- `#define XYSSL_ERR_NET_UNKNOWN_HOST -0x0F00`
- `#define XYSSL_ERR_NET_SOCKET_FAILED -0x0F10`
- `#define XYSSL_ERR_NET_CONNECT_FAILED -0x0F20`
- `#define XYSSL_ERR_NET_BIND_FAILED -0x0F30`
- `#define XYSSL_ERR_NET_LISTEN_FAILED -0x0F40`
- `#define XYSSL_ERR_NET_ACCEPT_FAILED -0x0F50`
- `#define XYSSL_ERR_NET_RECV_FAILED -0x0F60`
- `#define XYSSL_ERR_NET_SEND_FAILED -0x0F70`
- `#define XYSSL_ERR_NET_CONN_RESET -0x0F80`
- `#define XYSSL_ERR_NET_TRY_AGAIN -0x0F90`

### Functions

- `int net_connect (int *fd, char *host, int port)`  
*Initiate a TCP connection with host:port.*
- `int net_bind (int *fd, char *bind_ip, int port)`  
*Create a listening socket on bind\_ip:port. If bind\_ip == NULL, all interfaces are binded.*
- `int net_accept (int bind_fd, int *client_fd, void *client_ip)`  
*Accept a connection from a remote client.*
- `int net_set_block (int fd)`  
*Set the socket blocking.*
- `int net_set_nonblock (int fd)`  
*Set the socket non-blocking.*
- `void net_usleep (unsigned long usec)`  
*Portable usleep helper.*
- `int net_recv (void *ctx, unsigned char *buf, int len)`  
*Read at most 'len' characters. len is updated to reflect the actual number of characters read.*
- `int net_send (void *ctx, unsigned char *buf, int len)`  
*Write at most 'len' characters. len is updated to reflect the number of characters \_not\_ written.*
- `void net_close (int fd)`  
*Gracefully shutdown the connection.*

## 13.105.1 Detailed Description

Definition in file [net.h](#).

## 13.105.2 Define Documentation

### 13.105.2.1 `#define XYSSL_ERR_NET_ACCEPT_FAILED -0x0F50`

Definition at line 12 of file net.h.

Referenced by `net_accept()`.

### 13.105.2.2 `#define XYSSL_ERR_NET_BIND_FAILED -0x0F30`

Definition at line 10 of file net.h.

Referenced by `net_bind()`.

### 13.105.2.3 `#define XYSSL_ERR_NET_CONN_RESET -0x0F80`

Definition at line 15 of file net.h.

Referenced by `main()`, `net_recv()`, `net_send()`, and `ssl_test()`.

### 13.105.2.4 `#define XYSSL_ERR_NET_CONNECT_FAILED -0x0F20`

Definition at line 9 of file net.h.

Referenced by `net_connect()`.

### 13.105.2.5 `#define XYSSL_ERR_NET_LISTEN_FAILED -0x0F40`

Definition at line 11 of file net.h.

Referenced by `net_bind()`.

### 13.105.2.6 `#define XYSSL_ERR_NET_RECV_FAILED -0x0F60`

Definition at line 13 of file net.h.

Referenced by `net_recv()`.

### 13.105.2.7 `#define XYSSL_ERR_NET_SEND_FAILED -0x0F70`

Definition at line 14 of file net.h.

Referenced by `net_send()`.

### 13.105.2.8 `#define XYSSL_ERR_NET_SOCKET_FAILED -0x0F10`

Definition at line 8 of file net.h.

Referenced by `net_bind()`, and `net_connect()`.

### 13.105.2.9 `#define XYSSL_ERR_NET_TRY_AGAIN -0x0F90`

Definition at line 16 of file `net.h`.

Referenced by `main()`, `net_accept()`, `net_recv()`, `net_send()`, and `ssl_test()`.

### 13.105.2.10 `#define XYSSL_ERR_NET_UNKNOWN_HOST -0x0F00`

Definition at line 7 of file `net.h`.

Referenced by `net_connect()`.

## 13.105.3 Function Documentation

### 13.105.3.1 `int net_accept (int bind_fd, int * client_fd, void * client_ip)`

Accept a connection from a remote client.

#### Returns:

0 if successful, `XYSSL_ERR_NET_ACCEPT_FAILED`, or `XYSSL_ERR_NET_WOULD_BLOCK` if `bind_fd` was set to non-blocking and `accept()` is blocking.

Definition at line 212 of file `net.c`.

References `int`, `net_is_blocking()`, `XYSSL_ERR_NET_ACCEPT_FAILED`, and `XYSSL_ERR_NET_TRY_AGAIN`.

Referenced by `main()`, and `ssl_test()`.

### 13.105.3.2 `int net_bind (int * fd, char * bind_ip, int port)`

Create a listening socket on `bind_ip:port`. If `bind_ip == NULL`, all interfaces are binded.

#### Returns:

0 if successful, or one of: `XYSSL_ERR_NET_SOCKET_FAILED`, `XYSSL_ERR_NET_BIND_FAILED`, `XYSSL_ERR_NET_LISTEN_FAILED`

Definition at line 126 of file `net.c`.

References `net_htons()`, `SOCKET_ERROR`, `XYSSL_ERR_NET_BIND_FAILED`, `XYSSL_ERR_NET_LISTEN_FAILED`, and `XYSSL_ERR_NET_SOCKET_FAILED`.

Referenced by `main()`, and `ssl_test()`.

### 13.105.3.3 `void net_close (int fd)`

Gracefully shutdown the connection.

Definition at line 338 of file `net.c`.

Referenced by `main()`, and `ssl_test()`.



### 13.105.3.4 `int net_connect (int *fd, char *host, int port)`

Initiate a TCP connection with host:port.

#### Returns:

0 if successful, or one of: XYSSL\_ERR\_NET\_SOCKET\_FAILED, XYSSL\_ERR\_NET\_UNKNOWN\_HOST, XYSSL\_ERR\_NET\_CONNECT\_FAILED

Definition at line 81 of file net.c.

References net\_htons(), SOCKET\_ERROR, XYSSL\_ERR\_NET\_CONNECT\_FAILED, XYSSL\_ERR\_NET\_SOCKET\_FAILED, and XYSSL\_ERR\_NET\_UNKNOWN\_HOST.

Referenced by main(), and ssl\_test().

### 13.105.3.5 `int net_recv (void *ctx, unsigned char *buf, int len)`

Read at most 'len' characters. len is updated to reflect the actual number of characters read.

#### Returns:

This function returns the number of bytes received, or a negative error code; XYSSL\_ERR\_NET\_TRY\_AGAIN indicates read() is blocking.

Definition at line 277 of file net.c.

References net\_is\_blocking(), XYSSL\_ERR\_NET\_CONN\_RESET, XYSSL\_ERR\_NET\_RECV\_FAILED, and XYSSL\_ERR\_NET\_TRY\_AGAIN.

Referenced by main(), and ssl\_test().

### 13.105.3.6 `int net_send (void *ctx, unsigned char *buf, int len)`

Write at most 'len' characters. len is updated to reflect the number of characters \_not\_ written.

#### Returns:

This function returns the number of bytes sent, or a negative error code; XYSSL\_ERR\_NET\_TRY\_AGAIN indicates write() is blocking.

Definition at line 309 of file net.c.

References net\_is\_blocking(), XYSSL\_ERR\_NET\_CONN\_RESET, XYSSL\_ERR\_NET\_SEND\_FAILED, and XYSSL\_ERR\_NET\_TRY\_AGAIN.

Referenced by main(), and ssl\_test().

### 13.105.3.7 `int net_set_block (int fd)`

Set the socket blocking.

#### Returns:

0 if successful, or a non-zero error code

Definition at line 243 of file net.c.

**13.105.3.8** `int net_set_nonblock (int fd)`

Set the socket non-blocking.

**Returns:**

0 if successful, or a non-zero error code

Definition at line 253 of file net.c.

Referenced by `ssl_test()`.

**13.105.3.9** `void net_usleep (unsigned long usec)`

Portable usleep helper.

**Note:**

Real amount of time slept will not be less than `select()`'s timeout granularity (typically, 10ms).

Definition at line 266 of file net.c.

## v1.10.10/src/security/xyssl-0.9/include/xyssl/openssl.h File Reference

```
#include "xyssl/aes.h"
#include "xyssl/md5.h"
#include "xyssl/rsa.h"
#include "xyssl/sha1.h"
```

### Defines

- #define [AES\\_SIZE](#) 16
- #define [AES\\_BLOCK\\_SIZE](#) 16
- #define [AES\\_KEY](#) aes\_context
- #define [MD5\\_CTX](#) md5\_context
- #define [SHA\\_CTX](#) sha1\_context
- #define [SHA1\\_Init](#)(CTX) sha1\_starts( (CTX) )
- #define [SHA1\\_Update](#)(CTX, BUF, LEN) sha1\_update( (CTX), (unsigned char \*) (BUF), (LEN) )
- #define [SHA1\\_Final](#)(OUT, CTX) sha1\_finish( (CTX), (OUT) )
- #define [MD5\\_Init](#)(CTX) md5\_starts( (CTX) )
- #define [MD5\\_Update](#)(CTX, BUF, LEN) md5\_update( (CTX), (unsigned char \*) (BUF), (LEN) )
- #define [MD5\\_Final](#)(OUT, CTX) md5\_finish( (CTX), (OUT) )
- #define [AES\\_set\\_encrypt\\_key](#)(KEY, KEYSIZE, CTX) aes\_setkey\_enc( (CTX), (KEY), (KEYSIZE) )
- #define [AES\\_set\\_decrypt\\_key](#)(KEY, KEYSIZE, CTX) aes\_setkey\_dec( (CTX), (KEY), (KEYSIZE) )
- #define [AES\\_cbc\\_encrypt](#)(INPUT, OUTPUT, LEN, CTX, IV, MODE) aes\_crypt\_cbc( (CTX), (MODE), (LEN), (IV), (INPUT), (OUTPUT) )
- #define [RSA](#) rsa\_context
- #define [RSA\\_PKCS1\\_PADDING](#) 1
- #define [RSA\\_size](#)(CTX) (CTX)->len
- #define [RSA\\_free](#)(CTX) rsa\_free( CTX )
- #define [ERR\\_get\\_error](#)() "ERR\_get\_error() not supported"
- #define [RSA\\_blinding\\_off](#)(IGNORE)
- #define [d2i\\_RSAPrivateKey](#)(a, b, c) new\_rsa\_context

### Functions

- [int \\_\\_RSA\\_Passthrough](#) (void \*output, void \*input, [int size](#))
- [rsa\\_context \\* d2i\\_RSA\\_PUBKEY](#) (void \*ignore, unsigned char \*\*bufptr, [int len](#))
- [int RSA\\_public\\_decrypt](#) ([int size](#), unsigned char \*input, unsigned char \*output, RSA \*key, [int ignore](#))
- [int RSA\\_private\\_decrypt](#) ([int size](#), unsigned char \*input, unsigned char \*output, RSA \*key, [int ignore](#))
- [int RSA\\_public\\_encrypt](#) ([int size](#), unsigned char \*input, unsigned char \*output, RSA \*key, [int ignore](#))
- [int RSA\\_private\\_encrypt](#) ([int size](#), unsigned char \*input, unsigned char \*output, RSA \*key, [int ignore](#))

### 13.106.1 Detailed Description

Definition in file [openssl.h](#).

### 13.106.2 Define Documentation

#### 13.106.2.1 `#define AES_BLOCK_SIZE 16`

Definition at line 16 of file [openssl.h](#).

#### 13.106.2.2 `#define AES_cbc_encrypt(INPUT, OUTPUT, LEN, CTX, IV, MODE) aes_crypt_cbc( (CTX), (MODE), (LEN), (IV), (INPUT), (OUTPUT) )`

Definition at line 39 of file [openssl.h](#).

#### 13.106.2.3 `#define AES_KEY aes_context`

Definition at line 17 of file [openssl.h](#).

#### 13.106.2.4 `#define AES_set_decrypt_key(KEY, KEYSIZE, CTX) aes_setkey_dec( (CTX), (KEY), (KEYSIZE) )`

Definition at line 37 of file [openssl.h](#).

#### 13.106.2.5 `#define AES_set_encrypt_key(KEY, KEYSIZE, CTX) aes_setkey_enc( (CTX), (KEY), (KEYSIZE) )`

Definition at line 35 of file [openssl.h](#).

#### 13.106.2.6 `#define AES_SIZE 16`

Definition at line 15 of file [openssl.h](#).

#### 13.106.2.7 `#define d2i_RSAPrivateKey(a, b, c) new rsa_context`

Definition at line 102 of file [openssl.h](#).

#### 13.106.2.8 `#define ERR_get_error() "ERR_get_error() not supported"`

Definition at line 99 of file [openssl.h](#).

#### 13.106.2.9 `#define MD5_CTX md5_context`

Definition at line 18 of file [openssl.h](#).

### 13.106

/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/openssl.h

#### File Reference

705

**13.106.2.10** **#define MD5\_Final(OUT, CTX) md5\_finish( (CTX), (OUT) )**

Definition at line 32 of file openssl.h.

**13.106.2.11** **#define MD5\_Init(CTX) md5\_starts( (CTX) )**

Definition at line 28 of file openssl.h.

**13.106.2.12** **#define MD5\_Update(CTX, BUF, LEN) md5\_update( (CTX), (unsigned char \*) (BUF), (LEN) )**

Definition at line 30 of file openssl.h.

**13.106.2.13** **#define RSA\_rsa\_context**

Definition at line 95 of file openssl.h.

**13.106.2.14** **#define RSA\_blinding\_off(IGNORE)**

Definition at line 100 of file openssl.h.

**13.106.2.15** **#define RSA\_free(CTX) rsa\_free( CTX )**

Definition at line 98 of file openssl.h.

**13.106.2.16** **#define RSA\_PKCS1\_PADDING 1**

Definition at line 96 of file openssl.h.

**13.106.2.17** **#define RSA\_size(CTX) (CTX)->len**

Definition at line 97 of file openssl.h.

Referenced by RSA\_private\_encrypt(), and RSA\_public\_encrypt().

**13.106.2.18** **#define SHA1\_Final(OUT, CTX) sha1\_finish( (CTX), (OUT) )**

Definition at line 25 of file openssl.h.

**13.106.2.19** **#define SHA1\_Init(CTX) sha1\_starts( (CTX) )**

Definition at line 21 of file openssl.h.

**13.106.2.20** **#define SHA1\_Update(CTX, BUF, LEN) sha1\_update( (CTX), (unsigned char \*) (BUF), (LEN) )**

Definition at line 23 of file openssl.h.

**13.106.2.21 #define SHA\_CTX sha1\_context**

Definition at line 19 of file openssl.h.

**13.106.3 Function Documentation****13.106.3.1 int \_\_RSA\_Passthrough (void \* *output*, void \* *input*, int *size*) [inline]**

Definition at line 45 of file openssl.h.

**13.106.3.2 rsa\_context\* d2i\_RSA\_PUBKEY (void \* *ignore*, unsigned char \*\* *bufptr*, int *len*) [inline]**

Definition at line 51 of file openssl.h.

References `rsa_context::E`, `rsa_context::len`, `mpi_msb()`, `mpi_read_binary()`, and `rsa_context::N`.

**13.106.3.3 int RSA\_private\_decrypt (int *size*, unsigned char \* *input*, unsigned char \* *output*, RSA \* *key*, int *ignore*) [inline]**

Definition at line 105 of file openssl.h.

References `rsa_pkcs1_decrypt()`, and `RSA_PRIVATE`.

**13.106.3.4 int RSA\_private\_encrypt (int *size*, unsigned char \* *input*, unsigned char \* *output*, RSA \* *key*, int *ignore*) [inline]**

Definition at line 107 of file openssl.h.

References `rsa_pkcs1_encrypt()`, `RSA_PRIVATE`, and `RSA_size`.

**13.106.3.5 int RSA\_public\_decrypt (int *size*, unsigned char \* *input*, unsigned char \* *output*, RSA \* *key*, int *ignore*) [inline]**

Definition at line 104 of file openssl.h.

References `rsa_pkcs1_decrypt()`, and `RSA_PUBLIC`.

**13.106.3.6 int RSA\_public\_encrypt (int *size*, unsigned char \* *input*, unsigned char \* *output*, RSA \* *key*, int *ignore*) [inline]**

Definition at line 106 of file openssl.h.

References `rsa_pkcs1_encrypt()`, `RSA_PUBLIC`, and `RSA_size`.

13.107

/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/padlock.h

File Reference

707

**13.107** **/home/dko/Projects/mobilec/tags/MobileC-**  
**v1.10.10/src/security/xyssl-0.9/include/xyssl/padlock.h** **File**  
**Reference**

```
#include "xyssl/aes.h"
```

### 13.107.1 Detailed Description

Definition in file [padlock.h](#).

## 13.108 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/rsa.h File Reference

```
#include "bignum.h"
```

### Data Structures

- struct [rsa\\_context](#)  
*RSA context structure.*

### Defines

- #define [XYSSL\\_ERR\\_RSA\\_BAD\\_INPUT\\_DATA](#) -0x0400
- #define [XYSSL\\_ERR\\_RSA\\_INVALID\\_PADDING](#) -0x0410
- #define [XYSSL\\_ERR\\_RSA\\_KEY\\_GEN\\_FAILED](#) -0x0420
- #define [XYSSL\\_ERR\\_RSA\\_KEY\\_CHECK\\_FAILED](#) -0x0430
- #define [XYSSL\\_ERR\\_RSA\\_PUBLIC\\_FAILED](#) -0x0440
- #define [XYSSL\\_ERR\\_RSA\\_PRIVATE\\_FAILED](#) -0x0450
- #define [XYSSL\\_ERR\\_RSA\\_VERIFY\\_FAILED](#) -0x0460
- #define [RSA\\_RAW](#) 0
- #define [RSA\\_MD2](#) 2
- #define [RSA\\_MD4](#) 3
- #define [RSA\\_MD5](#) 4
- #define [RSA\\_SHA1](#) 5
- #define [RSA\\_SHA256](#) 6
- #define [RSA\\_PUBLIC](#) 0
- #define [RSA\\_PRIVATE](#) 1
- #define [RSA\\_PKCS\\_V15](#) 0
- #define [RSA\\_PKCS\\_V21](#) 1
- #define [RSA\\_SIGN](#) 1
- #define [RSA\\_CRYPT](#) 2
- #define [ASN1\\_HASH\\_MD5](#)
- #define [ASN1\\_HASH\\_SHA1](#)

### Functions

- void [rsa\\_init](#) ([rsa\\_context](#) \*ctx, int padding, int hash\_id, int(\*f\_rng)(void \*), void \*p\_rng)  
*Initialize an RSA context.*
- int [rsa\\_gen\\_key](#) ([rsa\\_context](#) \*ctx, int nbits, int exponent)  
*Generate an RSA keypair.*
- int [rsa\\_check\\_pubkey](#) ([rsa\\_context](#) \*ctx)  
*Check a public RSA key.*
- int [rsa\\_check\\_privkey](#) ([rsa\\_context](#) \*ctx)



*Check a private RSA key.*

- [int rsa\\_public](#) ([rsa\\_context](#) \*ctx, unsigned char \*input, unsigned char \*output)

*Do an RSA public key operation.*

- [int rsa\\_private](#) ([rsa\\_context](#) \*ctx, unsigned char \*input, unsigned char \*output)

*Do an RSA private key operation.*

- [int rsa\\_pkcs1\\_encrypt](#) ([rsa\\_context](#) \*ctx, [int](#) mode, [int](#) ilen, unsigned char \*input, unsigned char \*output)

*Add the message padding, then do an RSA operation.*

- [int rsa\\_pkcs1\\_decrypt](#) ([rsa\\_context](#) \*ctx, [int](#) mode, [int](#) \*olen, unsigned char \*input, unsigned char \*output)

*Do an RSA operation, then remove the message padding.*

- [int rsa\\_pkcs1\\_sign](#) ([rsa\\_context](#) \*ctx, [int](#) mode, [int](#) hash\_id, [int](#) hashlen, unsigned char \*hash, unsigned char \*sig)

*Do a private RSA to sign a message digest.*

- [int rsa\\_pkcs1\\_verify](#) ([rsa\\_context](#) \*ctx, [int](#) mode, [int](#) hash\_id, [int](#) hashlen, unsigned char \*hash, unsigned char \*sig)

*Do a public RSA and check the message digest.*

- void [rsa\\_free](#) ([rsa\\_context](#) \*ctx)

*Free the components of an RSA key.*

- [int rsa\\_self\\_test](#) ([int](#) verbose)

*Checkup routine.*

## 13.108.1 Detailed Description

Definition in file [rsa.h](#).

## 13.108.2 Define Documentation

### 13.108.2.1 #define ASN1\_HASH\_MDX

**Value:**

```
"\x30\x20\x30\x0C\x06\x08\x2A\x86\x48" \
"\x86\xF7\x0D\x02\x00\x05\x00\x04\x10"
```

Definition at line 45 of file [rsa.h](#).

Referenced by [rsa\\_pkcs1\\_sign\(\)](#), and [rsa\\_pkcs1\\_verify\(\)](#).

**13.108.2.2 #define ASN1\_HASH\_SHA1****Value:**

```
"\x30\x21\x30\x09\x06\x05\x2B\x0E\x03" \
"\x02\x1A\x05\x00\x04\x14"
```

Definition at line 49 of file rsa.h.

Referenced by rsa\_pkcs1\_sign(), and rsa\_pkcs1\_verify().

**13.108.2.3 #define RSA\_CRYPT 2**

Definition at line 34 of file rsa.h.

Referenced by rsa\_pkcs1\_decrypt(), and rsa\_pkcs1\_encrypt().

**13.108.2.4 #define RSA\_MD2 2**

Definition at line 21 of file rsa.h.

Referenced by rsa\_pkcs1\_sign(), rsa\_pkcs1\_verify(), x509\_hash(), and x509parse\_cert\_info().

**13.108.2.5 #define RSA\_MD4 3**

Definition at line 22 of file rsa.h.

Referenced by rsa\_pkcs1\_sign(), rsa\_pkcs1\_verify(), x509\_hash(), and x509parse\_cert\_info().

**13.108.2.6 #define RSA\_MD5 4**

Definition at line 23 of file rsa.h.

Referenced by rsa\_pkcs1\_sign(), rsa\_pkcs1\_verify(), x509\_hash(), and x509parse\_cert\_info().

**13.108.2.7 #define RSA\_PKCS\_V15 0**

Definition at line 30 of file rsa.h.

Referenced by generate\_RSA\_keys\_ciphertext(), generate\_RSA\_keys\_plaintext(), main(), rsa\_decryption(), rsa\_encryption(), rsa\_pkcs1\_decrypt(), rsa\_pkcs1\_encrypt(), rsa\_pkcs1\_sign(), and rsa\_pkcs1\_verify().

**13.108.2.8 #define RSA\_PKCS\_V21 1**

Definition at line 31 of file rsa.h.

**13.108.2.9 #define RSA\_PRIVATE 1**

Definition at line 28 of file rsa.h.

Referenced by main(), RSA\_private\_decrypt(), RSA\_private\_encrypt(), rsa\_self\_test(), ssl\_parse\_client\_key\_exchange(), ssl\_write\_certificate\_verify(), and ssl\_write\_server\_key\_exchange().

#### 13.108.2.10 **#define RSA\_PUBLIC 0**

Definition at line 27 of file rsa.h.

Referenced by main(), rsa\_pkcs1\_decrypt(), rsa\_pkcs1\_encrypt(), rsa\_pkcs1\_sign(), rsa\_pkcs1\_verify(), RSA\_public\_decrypt(), RSA\_public\_encrypt(), rsa\_self\_test(), ssl\_parse\_certificate\_verify(), ssl\_parse\_server\_key\_exchange(), ssl\_write\_client\_key\_exchange(), and x509parse\_verify().

#### 13.108.2.11 **#define RSA\_RAW 0**

Definition at line 20 of file rsa.h.

Referenced by rsa\_pkcs1\_sign(), rsa\_pkcs1\_verify(), ssl\_parse\_certificate\_verify(), ssl\_parse\_server\_key\_exchange(), ssl\_write\_certificate\_verify(), and ssl\_write\_server\_key\_exchange().

#### 13.108.2.12 **#define RSA\_SHA1 5**

Definition at line 24 of file rsa.h.

Referenced by main(), rsa\_pkcs1\_sign(), rsa\_pkcs1\_verify(), rsa\_self\_test(), x509\_hash(), and x509parse\_cert\_info().

#### 13.108.2.13 **#define RSA\_SHA256 6**

Definition at line 25 of file rsa.h.

#### 13.108.2.14 **#define RSA\_SIGN 1**

Definition at line 33 of file rsa.h.

Referenced by rsa\_pkcs1\_sign(), and rsa\_pkcs1\_verify().

#### 13.108.2.15 **#define XYSSL\_ERR\_RSA\_BAD\_INPUT\_DATA -0x0400**

Definition at line 9 of file rsa.h.

Referenced by rsa\_gen\_key(), rsa\_pkcs1\_decrypt(), rsa\_pkcs1\_encrypt(), rsa\_pkcs1\_sign(), rsa\_pkcs1\_verify(), rsa\_private(), and rsa\_public().

#### 13.108.2.16 **#define XYSSL\_ERR\_RSA\_INVALID\_PADDING -0x0410**

Definition at line 10 of file rsa.h.

Referenced by rsa\_pkcs1\_decrypt(), rsa\_pkcs1\_encrypt(), rsa\_pkcs1\_sign(), and rsa\_pkcs1\_verify().

#### 13.108.2.17 **#define XYSSL\_ERR\_RSA\_KEY\_CHECK\_FAILED -0x0430**

Definition at line 12 of file rsa.h.

Referenced by rsa\_check\_privkey(), and rsa\_check\_pubkey().

**13.108.2.18 #define XYSSL\_ERR\_RSA\_KEY\_GEN\_FAILED -0x0420**

Definition at line 11 of file rsa.h.

Referenced by rsa\_gen\_key().

**13.108.2.19 #define XYSSL\_ERR\_RSA\_PRIVATE\_FAILED -0x0450**

Definition at line 14 of file rsa.h.

Referenced by rsa\_private().

**13.108.2.20 #define XYSSL\_ERR\_RSA\_PUBLIC\_FAILED -0x0440**

Definition at line 13 of file rsa.h.

Referenced by rsa\_public().

**13.108.2.21 #define XYSSL\_ERR\_RSA\_VERIFY\_FAILED -0x0460**

Definition at line 15 of file rsa.h.

Referenced by rsa\_pkcs1\_verify().

**13.108.3 Function Documentation****13.108.3.1 int rsa\_check\_privkey (rsa\_context \* ctx)**

Check a private RSA key.

**Parameters:**

*ctx* RSA context to be checked

**Returns:**

0 if successful, or an XYSSL\_ERR\_RSA\_XXX error code

Definition at line 152 of file rsa.c.

References rsa\_context::D, rsa\_context::E, MPI\_CHK, mpi\_cmp\_int(), mpi\_cmp\_mpi(), mpi\_free(), mpi\_gcd(), mpi\_init(), mpi\_mod\_mpi(), mpi\_mul\_mpi(), mpi\_sub\_int(), rsa\_context::N, rsa\_context::P, rsa\_context::Q, rsa\_check\_pubkey(), and XYSSL\_ERR\_RSA\_KEY\_CHECK\_FAILED.

Referenced by rsa\_decryption(), rsa\_self\_test(), and x509parse\_key().

**13.108.3.2 int rsa\_check\_pubkey (rsa\_context \* ctx)**

Check a public RSA key.

**Parameters:**

*ctx* RSA context to be checked

**Returns:**

0 if successful, or an XYSSL\_ERR\_RSA\_XXX error code

Definition at line 132 of file rsa.c.

References `rsa_context::E`, `mpi_msb()`, `rsa_context::N`, `mpi::p`, and `XYSSL_ERR_RSA_KEY_CHECK_FAILED`.

Referenced by `rsa_check_privkey()`, `rsa_encryption()`, `rsa_self_test()`, and `x509parse_crt()`.

**13.108.3.3 void rsa\_free (rsa\_context \* ctx)**

Free the components of an RSA key.

Definition at line 558 of file rsa.c.

References `rsa_context::D`, `rsa_context::DP`, `rsa_context::DQ`, `rsa_context::E`, `mpi_free()`, `rsa_context::N`, `rsa_context::P`, `rsa_context::Q`, `rsa_context::QP`, `rsa_context::RN`, `rsa_context::RP`, and `rsa_context::RQ`.

Referenced by `main()`, `rsa_gen_key()`, `rsa_self_test()`, `ssl_test()`, `x509_free()`, `x509_self_test()`, and `x509parse_key()`.

**13.108.3.4 int rsa\_gen\_key (rsa\_context \* ctx, int nbits, int exponent)**

Generate an RSA keypair.

**Parameters:**

*ctx* RSA context that will hold the key

*nbits* size of the public key in bits

*exponent* public exponent (e.g., 65537)

**Note:**

[rsa\\_init\(\)](#) must be called beforehand to setup the RSA context (especially `f_rng` and `p_rng`).

**Returns:**

0 if successful, or an XYSSL\_ERR\_RSA\_XXX error code

Definition at line 60 of file rsa.c.

References `rsa_context::D`, `rsa_context::DP`, `rsa_context::DQ`, `rsa_context::E`, `rsa_context::f_rng`, `rsa_context::len`, `MPI_CHK`, `mpi_cmp_int()`, `mpi_cmp_mpi()`, `mpi_free()`, `mpi_gcd()`, `mpi_gen_prime()`, `mpi_init()`, `mpi_inv_mod()`, `mpi_lset()`, `mpi_mod_mpi()`, `mpi_msb()`, `mpi_mul_mpi()`, `mpi_sub_int()`, `mpi_swap()`, `rsa_context::N`, `rsa_context::P`, `rsa_context::p_rng`, `rsa_context::Q`, `rsa_context::QP`, `rsa_free()`, `XYSSL_ERR_RSA_BAD_INPUT_DATA`, and `XYSSL_ERR_RSA_KEY_GEN_FAILED`.

Referenced by `generate_RSA_keys_ciphertext()`, `generate_RSA_keys_plaintext()`, and `main()`.

**13.108.3.5 void rsa\_init (rsa\_context \* ctx, int padding, int hash\_id, int(\*) (void \*) f\_rng, void \* p\_rng)**

Initialize an RSA context.

**Parameters:**

*ctx* RSA context to be initialized  
*padding* RSA\_PKCS\_V15 or RSA\_PKCS\_V21  
*hash\_id* RSA\_PKCS\_V21 hash identifier  
*f\_rng* RNG function  
*p\_rng* RNG parameter

**Note:**

The *hash\_id* parameter is actually ignored when using RSA\_PKCS\_V15 padding.  
 Currently (xyssl-0.8), RSA\_PKCS\_V21 padding is not supported.

Definition at line 40 of file `rsa.c`.

References `rsa_context::f_rng`, `rsa_context::hash_id`, `rsa_context::p_rng`, and `rsa_context::padding`.

Referenced by `generate_RSA_keys_ciphertext()`, `generate_RSA_keys_plaintext()`, `main()`, `rsa_decryption()`, and `rsa_encryption()`.

### 13.108.3.6 `int rsa_pkcs1_decrypt(rsa_context * ctx, int mode, int * olen, unsigned char * input, unsigned char * output)`

Do an RSA operation, then remove the message padding.

**Parameters:**

*ctx* RSA context  
*mode* RSA\_PUBLIC or RSA\_PRIVATE  
*input* buffer holding the encrypted data  
*output* buffer that will hold the plaintext  
*olen* will contain the plaintext length

**Returns:**

0 if successful, or an XYSSL\_ERR\_RSA\_XXX error code

**Note:**

The output buffer must be as large as the size of `ctx->N` (eg. 128 bytes if RSA-1024 is used).

Definition at line 326 of file `rsa.c`.

References `buf`, `int`, `rsa_context::len`, `rsa_context::padding`, `RSA_CRYPT`, `RSA_PKCS_V15`, `rsa_private()`, `rsa_public()`, `RSA_PUBLIC`, `XYSSL_ERR_RSA_BAD_INPUT_DATA`, and `XYSSL_ERR_RSA_INVALID_PADDING`.

Referenced by `rsa_decryption()`, `RSA_private_decrypt()`, `RSA_public_decrypt()`, `rsa_self_test()`, and `ssl_parse_client_key_exchange()`.

### 13.108.3.7 `int rsa_pkcs1_encrypt(rsa_context * ctx, int mode, int ilen, unsigned char * input, unsigned char * output)`

Add the message padding, then do an RSA operation.

**Parameters:**

*ctx* RSA context  
*mode* RSA\_PUBLIC or RSA\_PRIVATE  
*ilen* contains the the plaintext length  
*input* buffer holding the data to be encrypted  
*output* buffer that will hold the ciphertext

**Returns:**

0 if successful, or an XYSSL\_ERR\_RSA\_XXX error code

**Note:**

The output buffer must be as large as the size of ctx->N (eg. 128 bytes if RSA-1024 is used).

Definition at line 280 of file rsa.c.

References `rsa_context::len`, `rsa_context::padding`, `RSA_CRYPT`, `RSA_PKCS_V15`, `rsa_private()`, `rsa_public()`, `RSA_PUBLIC`, `XYSSL_ERR_RSA_BAD_INPUT_DATA`, and `XYSSL_ERR_RSA_INVALID_PADDING`.

Referenced by `rsa_encryption()`, `RSA_private_encrypt()`, `RSA_public_encrypt()`, `rsa_self_test()`, and `ssl_write_client_key_exchange()`.

**13.108.3.8** `int rsa_pkcs1_sign (rsa_context * ctx, int mode, int hash_id, int hashlen, unsigned char * hash, unsigned char * sig)`

Do a private RSA to sign a message digest.

**Parameters:**

*ctx* RSA context  
*mode* RSA\_PUBLIC or RSA\_PRIVATE  
*hash\_id* RSA\_RAW, RSA\_MD{2,4,5} or RSA\_SHA{1,256}  
*hashlen* message digest length (for RSA\_RAW only)  
*hash* buffer holding the message digest  
*sig* buffer that will hold the ciphertext

**Returns:**

0 if the signing operation was successful, or an XYSSL\_ERR\_RSA\_XXX error code

**Note:**

The "sig" buffer must be as large as the size of ctx->N (eg. 128 bytes if RSA-1024 is used).

Definition at line 379 of file rsa.c.

References `ASN1_HASH_MD5`, `ASN1_HASH_SHA1`, `rsa_context::len`, `rsa_context::padding`, `RSA_MD2`, `RSA_MD4`, `RSA_MD5`, `RSA_PKCS_V15`, `rsa_private()`, `rsa_public()`, `RSA_PUBLIC`, `RSA_RAW`, `RSA_SHA1`, `RSA_SIGN`, `XYSSL_ERR_RSA_BAD_INPUT_DATA`, and `XYSSL_ERR_RSA_INVALID_PADDING`.

Referenced by `main()`, `rsa_self_test()`, `ssl_write_certificate_verify()`, and `ssl_write_server_key_exchange()`.

### 13.108.3.9 **int rsa\_pkcs1\_verify** (*rsa\_context* \* *ctx*, *int mode*, *int hash\_id*, *int hashlen*, *unsigned char* \* *hash*, *unsigned char* \* *sig*)

Do a public RSA and check the message digest.

#### Parameters:

*ctx* points to an RSA public key  
*mode* RSA\_PUBLIC or RSA\_PRIVATE  
*hash\_id* RSA\_RAW, RSA\_MD{2,4,5} or RSA\_SHA{1,256}  
*hashlen* message digest length (for RSA\_RAW only)  
*hash* buffer holding the message digest  
*sig* buffer holding the ciphertext

#### Returns:

0 if the verify operation was successful, or an XYSSL\_ERR\_RSA\_XXX error code

#### Note:

The "sig" buffer must be as large as the size of *ctx*->N (eg. 128 bytes if RSA-1024 is used).

Definition at line 468 of file rsa.c.

References ASN1\_HASH\_MDX, ASN1\_HASH\_SHA1, buf, int, rsa\_context::len, rsa\_context::padding, RSA\_MD2, RSA\_MD4, RSA\_MD5, RSA\_PKCS\_V15, rsa\_private(), rsa\_public(), RSA\_PUBLIC, RSA\_RAW, RSA\_SHA1, RSA\_SIGN, XYSSL\_ERR\_RSA\_BAD\_INPUT\_DATA, XYSSL\_ERR\_RSA\_INVALID\_PADDING, and XYSSL\_ERR\_RSA\_VERIFY\_FAILED.

Referenced by main(), rsa\_self\_test(), ssl\_parse\_certificate\_verify(), ssl\_parse\_server\_key\_exchange(), and x509parse\_verify().

### 13.108.3.10 **int rsa\_private** (*rsa\_context* \* *ctx*, *unsigned char* \* *input*, *unsigned char* \* *output*)

Do an RSA private key operation.

#### Parameters:

*ctx* RSA context  
*input* input buffer  
*output* output buffer

#### Returns:

0 if successful, or an XYSSL\_ERR\_RSA\_XXX error code

#### Note:

The input and output buffers must be large enough (eg. 128 bytes if RSA-1024 is used).

Definition at line 221 of file rsa.c.

References rsa\_context::D, rsa\_context::DP, rsa\_context::DQ, rsa\_context::len, mpi\_add\_mpi(), MPI\_CHK, mpi\_cmp\_mpi(), mpi\_exp\_mod(), mpi\_free(), mpi\_init(), mpi\_mod\_mpi(), mpi\_mul\_mpi(), mpi\_read\_binary(), mpi\_sub\_mpi(), mpi\_write\_binary(), rsa\_context::N, rsa\_context::P, rsa\_context::Q, rsa\_context::QP, rsa\_context::RN, rsa\_context::RP, rsa\_context::RQ, XYSSL\_ERR\_RSA\_BAD\_INPUT\_DATA, and XYSSL\_ERR\_RSA\_PRIVATE\_FAILED.



Referenced by `main()`, `rsa_pkcs1_decrypt()`, `rsa_pkcs1_encrypt()`, `rsa_pkcs1_sign()`, and `rsa_pkcs1_verify()`.

### 13.108.3.11 `int rsa_public (rsa_context * ctx, unsigned char * input, unsigned char * output)`

Do an RSA public key operation.

#### Parameters:

*ctx* RSA context  
*input* input buffer  
*output* output buffer

#### Returns:

0 if successful, or an `XYSSL_ERR_RSA_XXX` error code

#### Note:

This function does NOT take care of message padding. Also, be sure to set `input[0] = 0`.  
The input and output buffers must be large enough (eg. 128 bytes if RSA-1024 is used).

Definition at line 187 of file `rsa.c`.

References `rsa_context::E`, `rsa_context::len`, `MPI_CHK`, `mpi_cmp_mpi()`, `mpi_exp_mod()`, `mpi_free()`, `mpi_init()`, `mpi_read_binary()`, `mpi_write_binary()`, `rsa_context::N`, `rsa_context::RN`, `XYSSL_ERR_RSA_BAD_INPUT_DATA`, and `XYSSL_ERR_RSA_PUBLIC_FAILED`.

Referenced by `main()`, `rsa_pkcs1_decrypt()`, `rsa_pkcs1_encrypt()`, `rsa_pkcs1_sign()`, and `rsa_pkcs1_verify()`.

### 13.108.3.12 `int rsa_self_test (int verbose)`

Checkup routine.

#### Returns:

0 if successful, or 1 if the test failed

Definition at line 627 of file `rsa.c`.

References `rsa_context::D`, `rsa_context::DP`, `rsa_context::DQ`, `rsa_context::E`, `KEY_LEN`, `rsa_context::len`, `mpi_read_string()`, `rsa_context::N`, `rsa_context::P`, `PT_LEN`, `rsa_context::Q`, `rsa_context::QP`, `rsa_check_privkey()`, `rsa_check_pubkey()`, `RSA_D`, `RSA_DP`, `RSA_DQ`, `RSA_E`, `rsa_free()`, `RSA_N`, `RSA_P`, `rsa_pkcs1_decrypt()`, `rsa_pkcs1_encrypt()`, `rsa_pkcs1_sign()`, `rsa_pkcs1_verify()`, `RSA_PRIVATE`, `RSA_PT`, `RSA_PUBLIC`, `RSA_Q`, `RSA_QP`, `RSA_SHA1`, and `sha1()`.

Referenced by `main()`.

## 13.109 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/sha1.h File Reference

### Data Structures

- struct [sha1\\_context](#)  
*SHA-1 context structure.*

### Functions

- void [sha1\\_starts](#) ([sha1\\_context](#) \*ctx)  
*SHA-1 context setup.*
- void [sha1\\_update](#) ([sha1\\_context](#) \*ctx, unsigned char \*input, int ilen)  
*SHA-1 process buffer.*
- void [sha1\\_finish](#) ([sha1\\_context](#) \*ctx, unsigned char output[20])  
*SHA-1 final digest.*
- void [sha1](#) (unsigned char \*input, int ilen, unsigned char output[20])  
*Output = SHA-1( input buffer ).*
- int [sha1\\_file](#) (char \*path, unsigned char output[20])  
*Output = SHA-1( file contents ).*
- void [sha1\\_hmac\\_starts](#) ([sha1\\_context](#) \*ctx, unsigned char \*key, int keylen)  
*SHA-1 HMAC context setup.*
- void [sha1\\_hmac\\_update](#) ([sha1\\_context](#) \*ctx, unsigned char \*input, int ilen)  
*SHA-1 HMAC process buffer.*
- void [sha1\\_hmac\\_finish](#) ([sha1\\_context](#) \*ctx, unsigned char output[20])  
*SHA-1 HMAC final digest.*
- void [sha1\\_hmac](#) (unsigned char \*key, int keylen, unsigned char \*input, int ilen, unsigned char output[20])  
*Output = HMAC-SHA-1( hmac key, input buffer ).*
- int [sha1\\_self\\_test](#) (int verbose)  
*Checkup routine.*

### 13.109.1 Detailed Description

Definition in file [sha1.h](#).

---

**13.109.2 Function Documentation**

---

**13.109.2.1 void sha1 (unsigned char \* *input*, int *ilen*, unsigned char *output*[20])**

Output = SHA-1( *input* buffer ).

**Parameters:**

*input* buffer holding the data  
*ilen* length of the input data  
*output* SHA-1 checksum result

Definition at line 313 of file sha1.c.

References sha1\_finish(), sha1\_starts(), and sha1\_update().

Referenced by main(), rsa\_self\_test(), sha1\_hmac\_starts(), ssl\_calc\_verify(), ssl\_derive\_keys(), ssl\_mac\_sha1(), ssl\_parse\_finished(), ssl\_parse\_server\_key\_exchange(), ssl\_write\_finished(), ssl\_write\_server\_key\_exchange(), and x509\_hash().

**13.109.2.2 int sha1\_file (char \* *path*, unsigned char *output*[20])**

Output = SHA-1( *file* contents ).

**Parameters:**

*path* input file name  
*output* SHA-1 checksum result

**Returns:**

0 if successful, 1 if fopen failed, or 2 if fread failed

Definition at line 327 of file sha1.c.

References buf, f, sha1\_finish(), sha1\_starts(), and sha1\_update().

Referenced by main(), and sha1\_wrapper().

**13.109.2.3 void sha1\_finish (sha1\_context \* *ctx*, unsigned char *output*[20])**

SHA-1 final digest.

**Parameters:**

*ctx* SHA-1 context  
*output* SHA-1 checksum result

Definition at line 284 of file sha1.c.

References PUT\_ULONG\_BE, sha1\_padding, sha1\_update(), sha1\_context::state, and sha1\_context::total.

Referenced by sha1(), sha1\_file(), sha1\_hmac\_finish(), sha1\_self\_test(), ssl\_calc\_finished(), ssl\_calc\_verify(), ssl\_derive\_keys(), ssl\_mac\_sha1(), ssl\_parse\_server\_key\_exchange(), and ssl\_write\_server\_key\_exchange().

**13.109.2.4 void sha1\_hmac (unsigned char \* *key*, int *keylen*, unsigned char \* *input*, int *ilen*, unsigned char *output*[20])**

Output = HMAC-SHA-1( hmac key, input buffer ).

**Parameters:**

*key* HMAC secret key  
*keylen* length of the HMAC key  
*input* buffer holding the data  
*ilen* length of the input data  
*output* HMAC-SHA-1 result

Definition at line 413 of file sha1.c.

References sha1\_hmac\_finish(), sha1\_hmac\_starts(), and sha1\_hmac\_update().

Referenced by ssl\_decrypt\_buf(), ssl\_encrypt\_buf(), and tls1\_prf().

**13.109.2.5 void sha1\_hmac\_finish (sha1\_context \* *ctx*, unsigned char *output*[20])**

SHA-1 HMAC final digest.

**Parameters:**

*ctx* HMAC context  
*output* SHA-1 HMAC checksum result

Definition at line 397 of file sha1.c.

References sha1\_context::opad, sha1\_finish(), sha1\_starts(), and sha1\_update().

Referenced by sha1\_hmac(), and sha1\_self\_test().

**13.109.2.6 void sha1\_hmac\_starts (sha1\_context \* *ctx*, unsigned char \* *key*, int *keylen*)**

SHA-1 HMAC context setup.

**Parameters:**

*ctx* HMAC context to be initialized  
*key* HMAC secret key  
*keylen* length of the HMAC key

Definition at line 359 of file sha1.c.

References sha1\_context::ipad, sha1\_context::opad, sha1(), sha1\_starts(), and sha1\_update().

Referenced by sha1\_hmac(), and sha1\_self\_test().

**13.109.2.7 void sha1\_hmac\_update (sha1\_context \* *ctx*, unsigned char \* *input*, int *ilen*)**

SHA-1 HMAC process buffer.

**Parameters:***ctx* HMAC context*input* buffer holding the data*ilen* length of the input data

Definition at line 389 of file sha1.c.

References sha1\_update().

Referenced by sha1\_hmac(), and sha1\_self\_test().

**13.109.2.8 int sha1\_self\_test (int verbose)**

Checksum routine.

**Returns:**

0 if successful, or 1 if the test failed

Definition at line 521 of file sha1.c.

References buf, sha1\_finish(), sha1\_hmac\_finish(), sha1\_hmac\_starts(), sha1\_hmac\_test\_buf, sha1\_hmac\_test\_buflen, sha1\_hmac\_test\_key, sha1\_hmac\_test\_keylen, sha1\_hmac\_test\_sum, sha1\_hmac\_update(), sha1\_starts(), sha1\_test\_buf, sha1\_test\_buflen, sha1\_test\_sum, and sha1\_update().

Referenced by main().

**13.109.2.9 void sha1\_starts (sha1\_context \* ctx)**

SHA-1 context setup.

**Parameters:***ctx* context to be initialized

Definition at line 61 of file sha1.c.

References sha1\_context::state, and sha1\_context::total.

Referenced by sha1(), sha1\_file(), sha1\_hmac\_finish(), sha1\_hmac\_starts(), sha1\_self\_test(), ssl\_calc\_finished(), ssl\_calc\_verify(), ssl\_derive\_keys(), ssl\_init(), ssl\_mac\_sha1(), ssl\_parse\_server\_key\_exchange(), and ssl\_write\_server\_key\_exchange().

**13.109.2.10 void sha1\_update (sha1\_context \* ctx, unsigned char \* input, int ilen)**

SHA-1 process buffer.

**Parameters:***ctx* SHA-1 context*input* buffer holding the data*ilen* length of the input data

Definition at line 232 of file sha1.c.

References sha1\_context::buffer, sha1\_process(), and sha1\_context::total.

Referenced by sha1(), sha1\_file(), sha1\_finish(), sha1\_hmac\_finish(), sha1\_hmac\_starts(), sha1\_hmac\_update(), sha1\_self\_test(), ssl\_calc\_finished(), ssl\_calc\_verify(), ssl\_derive\_keys(), ssl\_mac\_sha1(), ssl\_parse\_client\_hello(), ssl\_parse\_server\_key\_exchange(), ssl\_read\_record(), ssl\_write\_record(), and ssl\_write\_server\_key\_exchange().

## 13.110 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/sha2.h File Reference

### Data Structures

- struct [sha2\\_context](#)  
*SHA-256 context structure.*

### Functions

- void [sha2\\_starts](#) ([sha2\\_context](#) \*ctx, [int](#) is224)  
*SHA-256 context setup.*
- void [sha2\\_update](#) ([sha2\\_context](#) \*ctx, unsigned char \*input, [int](#) ilen)  
*SHA-256 process buffer.*
- void [sha2\\_finish](#) ([sha2\\_context](#) \*ctx, unsigned char output[32])  
*SHA-256 final digest.*
- void [sha2](#) (unsigned char \*input, [int](#) ilen, unsigned char output[32], [int](#) is224)  
*Output = SHA-256( input buffer ).*
- [int](#) [sha2\\_file](#) (char \*path, unsigned char output[32], [int](#) is224)  
*Output = SHA-256( file contents ).*
- void [sha2\\_hmac\\_starts](#) ([sha2\\_context](#) \*ctx, unsigned char \*key, [int](#) keylen, [int](#) is224)  
*SHA-256 HMAC context setup.*
- void [sha2\\_hmac\\_update](#) ([sha2\\_context](#) \*ctx, unsigned char \*input, [int](#) ilen)  
*SHA-256 HMAC process buffer.*
- void [sha2\\_hmac\\_finish](#) ([sha2\\_context](#) \*ctx, unsigned char output[32])  
*SHA-256 HMAC final digest.*
- void [sha2\\_hmac](#) (unsigned char \*key, [int](#) keylen, unsigned char \*input, [int](#) ilen, unsigned char output[32], [int](#) is224)  
*Output = HMAC-SHA-256( hmac key, input buffer ).*
- [int](#) [sha2\\_self\\_test](#) ([int](#) verbose)  
*Checkup routine.*

### 13.110.1 Detailed Description

Definition in file [sha2.h](#).

### 13.110.2 Function Documentation

#### 13.110.2.1 void sha2 (unsigned char \* *input*, int *ilen*, unsigned char *output*[32], int *is224*)

Output = SHA-256( input buffer ).

**Parameters:**

*input* buffer holding the data

*ilen* length of the input data

*output* SHA-224/256 checksum result

*is224* 0 = use SHA256, 1 = use SHA224

Definition at line 314 of file sha2.c.

References sha2\_finish(), sha2\_starts(), and sha2\_update().

Referenced by main(), and sha2\_hmac\_starts().

#### 13.110.2.2 int sha2\_file (char \* *path*, unsigned char *output*[32], int *is224*)

Output = SHA-256( file contents ).

**Parameters:**

*path* input file name

*output* SHA-224/256 checksum result

*is224* 0 = use SHA256, 1 = use SHA224

**Returns:**

0 if successful, 1 if fopen failed, or 2 if fread failed

Definition at line 329 of file sha2.c.

References buf, f, sha2\_finish(), sha2\_starts(), and sha2\_update().

Referenced by sha2\_wrapper().

#### 13.110.2.3 void sha2\_finish (sha2\_context \* *ctx*, unsigned char *output*[32])

SHA-256 final digest.

**Parameters:**

*ctx* SHA-256 context

*output* SHA-224/256 checksum result

Definition at line 280 of file sha2.c.

References sha2\_context::is224, PUT\_ULONG\_BE, sha2\_padding, sha2\_update(), sha2\_context::state, and sha2\_context::total.

Referenced by aes\_en\_de(), main(), sha2(), sha2\_file(), sha2\_hmac\_finish(), and sha2\_self\_test().



### 13.110

/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/sha2.h File Reference 725

**13.110.2.4** `void sha2_hmac (unsigned char * key, int keylen, unsigned char * input, int ilen, unsigned char output[32], int is224)`

Output = HMAC-SHA-256( hmac key, input buffer ).

#### Parameters:

*key* HMAC secret key

*keylen* length of the HMAC key

*input* buffer holding the data

*ilen* length of the input data

*output* HMAC-SHA-224/256 result

*is224* 0 = use SHA256, 1 = use SHA224

Definition at line 420 of file sha2.c.

References sha2\_hmac\_finish(), sha2\_hmac\_starts(), and sha2\_hmac\_update().

**13.110.2.5** `void sha2_hmac_finish (sha2_context * ctx, unsigned char output[32])`

SHA-256 HMAC final digest.

#### Parameters:

*ctx* HMAC context

*output* SHA-224/256 HMAC checksum result

Definition at line 400 of file sha2.c.

References sha2\_context::is224, sha2\_context::opad, sha2\_finish(), sha2\_starts(), and sha2\_update().

Referenced by aes\_en\_de(), main(), sha2\_hmac(), and sha2\_self\_test().

**13.110.2.6** `void sha2_hmac_starts (sha2_context * ctx, unsigned char * key, int keylen, int is224)`

SHA-256 HMAC context setup.

#### Parameters:

*ctx* HMAC context to be initialized

*key* HMAC secret key

*keylen* length of the HMAC key

*is224* 0 = use SHA256, 1 = use SHA224

Definition at line 361 of file sha2.c.

References sha2\_context::ipad, sha2\_context::opad, sha2(), sha2\_starts(), and sha2\_update().

Referenced by aes\_en\_de(), main(), sha2\_hmac(), and sha2\_self\_test().

**13.110.2.7 void sha2\_hmac\_update (sha2\_context \* *ctx*, unsigned char \* *input*, int *ilen*)**

SHA-256 HMAC process buffer.

**Parameters:**

*ctx* HMAC context

*input* buffer holding the data

*ilen* length of the input data

Definition at line 392 of file sha2.c.

References sha2\_update().

Referenced by aes\_en\_de(), main(), sha2\_hmac(), and sha2\_self\_test().

**13.110.2.8 int sha2\_self\_test (int *verbose*)**

Checkup routine.

**Returns:**

0 if successful, or 1 if the test failed

Definition at line 599 of file sha2.c.

References buf, sha2\_finish(), sha2\_hmac\_finish(), sha2\_hmac\_starts(), sha2\_hmac\_test\_buf, sha2\_hmac\_test\_buflen, sha2\_hmac\_test\_key, sha2\_hmac\_test\_keylen, sha2\_hmac\_test\_sum, sha2\_hmac\_update(), sha2\_starts(), sha2\_test\_buf, sha2\_test\_buflen, sha2\_test\_sum, and sha2\_update().

Referenced by main().

**13.110.2.9 void sha2\_starts (sha2\_context \* *ctx*, int *is224*)**

SHA-256 context setup.

**Parameters:**

*ctx* context to be initialized

*is224* 0 = use SHA256, 1 = use SHA224

Definition at line 61 of file sha2.c.

References sha2\_context::is224, sha2\_context::state, and sha2\_context::total.

Referenced by aes\_en\_de(), main(), sha2(), sha2\_file(), sha2\_hmac\_finish(), sha2\_hmac\_starts(), and sha2\_self\_test().

**13.110.2.10 void sha2\_update (sha2\_context \* *ctx*, unsigned char \* *input*, int *ilen*)**

SHA-256 process buffer.

**Parameters:**

*ctx* SHA-256 context

*input* buffer holding the data

13.110

/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/sha2.h File

Reference

727

---

*ilen* length of the input data

Definition at line 228 of file sha2.c.

References sha2\_context::buffer, sha2\_process(), and sha2\_context::total.

Referenced by aes\_en\_de(), main(), sha2(), sha2\_file(), sha2\_finish(), sha2\_hmac\_finish(), sha2\_hmac\_starts(), sha2\_hmac\_update(), and sha2\_self\_test().

## 13.111 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/sha4.h File Reference

### Data Structures

- struct [sha4\\_context](#)  
*SHA-512 context structure.*

### Defines

- #define [UL64](#)(x) x##ULL
- #define [int64](#) long long

### Functions

- void [sha4\\_starts](#) ([sha4\\_context](#) \*ctx, [int](#) is384)  
*SHA-512 context setup.*
- void [sha4\\_update](#) ([sha4\\_context](#) \*ctx, unsigned char \*input, [int](#) ilen)  
*SHA-512 process buffer.*
- void [sha4\\_finish](#) ([sha4\\_context](#) \*ctx, unsigned char output[64])  
*SHA-512 final digest.*
- void [sha4](#) (unsigned char \*input, [int](#) ilen, unsigned char output[64], [int](#) is384)  
*Output = SHA-512( input buffer ).*
- [int](#) [sha4\\_file](#) (char \*path, unsigned char output[64], [int](#) is384)  
*Output = SHA-512( file contents ).*
- void [sha4\\_hmac\\_starts](#) ([sha4\\_context](#) \*ctx, unsigned char \*key, [int](#) keylen, [int](#) is384)  
*SHA-512 HMAC context setup.*
- void [sha4\\_hmac\\_update](#) ([sha4\\_context](#) \*ctx, unsigned char \*input, [int](#) ilen)  
*SHA-512 HMAC process buffer.*
- void [sha4\\_hmac\\_finish](#) ([sha4\\_context](#) \*ctx, unsigned char output[64])  
*SHA-512 HMAC final digest.*
- void [sha4\\_hmac](#) (unsigned char \*key, [int](#) keylen, unsigned char \*input, [int](#) ilen, unsigned char output[64], [int](#) is384)  
*Output = HMAC-SHA-512( hmac key, input buffer ).*
- [int](#) [sha4\\_self\\_test](#) ([int](#) verbose)  
*Checkup routine.*

**13.111.1 Detailed Description**

---

Definition in file [sha4.h](#).

**13.111.2 Define Documentation****13.111.2.1 #define int64 long long**

Definition at line 12 of file sha4.h.

Referenced by sha4\_finish(), sha4\_process(), and sha4\_update().

**13.111.2.2 #define UL64(x) x##ULL**

Definition at line 11 of file sha4.h.

Referenced by sha4\_starts().

**13.111.3 Function Documentation****13.111.3.1 void sha4 (unsigned char \* *input*, int *ilen*, unsigned char *output*[64], int *is384*)**

Output = SHA-512( input buffer ).

**Parameters:**

*input* buffer holding the data

*ilen* length of the input data

*output* SHA-384/512 checksum result

*is384* 0 = use SHA512, 1 = use SHA384

Definition at line 312 of file sha4.c.

References sha4\_finish(), sha4\_starts(), and sha4\_update().

Referenced by sha4\_hmac\_starts().

**13.111.3.2 int sha4\_file (char \* *path*, unsigned char *output*[64], int *is384*)**

Output = SHA-512( file contents ).

**Parameters:**

*path* input file name

*output* SHA-384/512 checksum result

*is384* 0 = use SHA512, 1 = use SHA384

**Returns:**

0 if successful, 1 if fopen failed, or 2 if fread failed

Definition at line 327 of file sha4.c.

References buf, f, sha4\_finish(), sha4\_starts(), and sha4\_update().

**13.111.3.3 void sha4\_finish (sha4\_context \* ctx, unsigned char output[64])**

SHA-512 final digest.

**Parameters:**

*ctx* SHA-512 context

*output* SHA-384/512 checksum result

Definition at line 276 of file sha4.c.

References int, int64, sha4\_context::is384, PUT\_UINT64\_BE, sha4\_padding, sha4\_update(), sha4\_context::state, and sha4\_context::total.

Referenced by sha4(), sha4\_file(), sha4\_hmac\_finish(), and sha4\_self\_test().

**13.111.3.4 void sha4\_hmac (unsigned char \* key, int keylen, unsigned char \* input, int ilen, unsigned char output[64], int is384)**

Output = HMAC-SHA-512( hmac key, input buffer ).

**Parameters:**

*key* HMAC secret key

*keylen* length of the HMAC key

*input* buffer holding the data

*ilen* length of the input data

*output* HMAC-SHA-384/512 result

*is384* 0 = use SHA512, 1 = use SHA384

Definition at line 419 of file sha4.c.

References sha4\_hmac\_finish(), sha4\_hmac\_starts(), and sha4\_hmac\_update().

**13.111.3.5 void sha4\_hmac\_finish (sha4\_context \* ctx, unsigned char output[64])**

SHA-512 HMAC final digest.

**Parameters:**

*ctx* HMAC context

*output* SHA-384/512 HMAC checksum result

Definition at line 399 of file sha4.c.

References sha4\_context::is384, sha4\_context::opad, sha4\_finish(), sha4\_starts(), and sha4\_update().

Referenced by sha4\_hmac(), and sha4\_self\_test().

**13.111.3.6 void sha4\_hmac\_starts (sha4\_context \* ctx, unsigned char \* key, int keylen, int is384)**

SHA-512 HMAC context setup.

**Parameters:**

*ctx* HMAC context to be initialized  
*is384* 0 = use SHA512, 1 = use SHA384  
*key* HMAC secret key  
*keylen* length of the HMAC key

Definition at line 359 of file sha4.c.

References sha4\_context::ipad, sha4\_context::opad, sha4(), sha4\_starts(), and sha4\_update().

Referenced by sha4\_hmac(), and sha4\_self\_test().

**13.111.3.7 void sha4\_hmac\_update (sha4\_context \* ctx, unsigned char \* input, int ilen)**

SHA-512 HMAC process buffer.

**Parameters:**

*ctx* HMAC context  
*input* buffer holding the data  
*ilen* length of the input data

Definition at line 390 of file sha4.c.

References sha4\_update().

Referenced by sha4\_hmac(), and sha4\_self\_test().

**13.111.3.8 int sha4\_self\_test (int verbose)**

Checkup routine.

**Returns:**

0 if successful, or 1 if the test failed

Definition at line 654 of file sha4.c.

References buf, sha4\_finish(), sha4\_hmac\_finish(), sha4\_hmac\_starts(), sha4\_hmac\_test\_buf, sha4\_hmac\_test\_buflen, sha4\_hmac\_test\_key, sha4\_hmac\_test\_keylen, sha4\_hmac\_test\_sum, sha4\_hmac\_update(), sha4\_starts(), sha4\_test\_buf, sha4\_test\_buflen, sha4\_test\_sum, and sha4\_update().

Referenced by main().

**13.111.3.9 void sha4\_starts (sha4\_context \* ctx, int is384)**

SHA-512 context setup.

**Parameters:**

*ctx* context to be initialized  
*is384* 0 = use SHA512, 1 = use SHA384

Definition at line 116 of file sha4.c.

References sha4\_context::is384, sha4\_context::state, sha4\_context::total, and UL64.

Referenced by sha4(), sha4\_file(), sha4\_hmac\_finish(), sha4\_hmac\_starts(), and sha4\_self\_test().

**13.111.3.10 void sha4\_update (sha4\_context \* *ctx*, unsigned char \* *input*, int *ilen*)**

SHA-512 process buffer.

**Parameters:**

*ctx* SHA-512 context

*input* buffer holding the data

*ilen* length of the input data

Definition at line 221 of file sha4.c.

References sha4\_context::buffer, int, int64, sha4\_process(), and sha4\_context::total.

Referenced by sha4(), sha4\_file(), sha4\_finish(), sha4\_hmac\_finish(), sha4\_hmac\_starts(), sha4\_hmac\_update(), and sha4\_self\_test().



## 13.112 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/ssl.h File Reference

```
#include <time.h>
#include "xyssl/net.h"
#include "xyssl/dhm.h"
#include "xyssl/rsa.h"
#include "xyssl/md5.h"
#include "xyssl/sha1.h"
#include "xyssl/x509.h"
```

### Data Structures

- struct [\\_ssl\\_session](#)
- struct [\\_ssl\\_context](#)

### Defines

- #define [XYSSL\\_ERR\\_SSL\\_FEATURE\\_UNAVAILABLE](#) -0x1000
- #define [XYSSL\\_ERR\\_SSL\\_BAD\\_INPUT\\_DATA](#) -0x1800
- #define [XYSSL\\_ERR\\_SSL\\_INVALID\\_MAC](#) -0x2000
- #define [XYSSL\\_ERR\\_SSL\\_INVALID\\_RECORD](#) -0x2800
- #define [XYSSL\\_ERR\\_SSL\\_INVALID\\_MODULUS\\_SIZE](#) -0x3000
- #define [XYSSL\\_ERR\\_SSL\\_UNKNOWN\\_CIPHER](#) -0x3800
- #define [XYSSL\\_ERR\\_SSL\\_NO\\_CIPHER\\_CHOSEN](#) -0x4000
- #define [XYSSL\\_ERR\\_SSL\\_NO\\_SESSION\\_FOUND](#) -0x4800
- #define [XYSSL\\_ERR\\_SSL\\_NO\\_CLIENT\\_CERTIFICATE](#) -0x5000
- #define [XYSSL\\_ERR\\_SSL\\_CERTIFICATE\\_TOO\\_LARGE](#) -0x5800
- #define [XYSSL\\_ERR\\_SSL\\_CERTIFICATE\\_REQUIRED](#) -0x6000
- #define [XYSSL\\_ERR\\_SSL\\_PRIVATE\\_KEY\\_REQUIRED](#) -0x6800
- #define [XYSSL\\_ERR\\_SSL\\_CA\\_CHAIN\\_REQUIRED](#) -0x7000
- #define [XYSSL\\_ERR\\_SSL\\_UNEXPECTED\\_MESSAGE](#) -0x7800
- #define [XYSSL\\_ERR\\_SSL\\_FATAL\\_ALERT\\_MESSAGE](#) -0x8000
- #define [XYSSL\\_ERR\\_SSL\\_PEER\\_VERIFY\\_FAILED](#) -0x8800
- #define [XYSSL\\_ERR\\_SSL\\_PEER\\_CLOSE\\_NOTIFY](#) -0x9000
- #define [XYSSL\\_ERR\\_SSL\\_BAD\\_HS\\_CLIENT\\_HELLO](#) -0x9800
- #define [XYSSL\\_ERR\\_SSL\\_BAD\\_HS\\_SERVER\\_HELLO](#) -0xA000
- #define [XYSSL\\_ERR\\_SSL\\_BAD\\_HS\\_CERTIFICATE](#) -0xA800
- #define [XYSSL\\_ERR\\_SSL\\_BAD\\_HS\\_CERTIFICATE\\_REQUEST](#) -0xB000
- #define [XYSSL\\_ERR\\_SSL\\_BAD\\_HS\\_SERVER\\_KEY\\_EXCHANGE](#) -0xB800
- #define [XYSSL\\_ERR\\_SSL\\_BAD\\_HS\\_SERVER\\_HELLO\\_DONE](#) -0xC000
- #define [XYSSL\\_ERR\\_SSL\\_BAD\\_HS\\_CLIENT\\_KEY\\_EXCHANGE](#) -0xC800
- #define [XYSSL\\_ERR\\_SSL\\_BAD\\_HS\\_CERTIFICATE\\_VERIFY](#) -0xD000
- #define [XYSSL\\_ERR\\_SSL\\_BAD\\_HS\\_CHANGE\\_CIPHER\\_SPEC](#) -0xD800
- #define [XYSSL\\_ERR\\_SSL\\_BAD\\_HS\\_FINISHED](#) -0xE000

- `#define SSL_MAJOR_VERSION_3 3`
- `#define SSL_MINOR_VERSION_0 0`
- `#define SSL_MINOR_VERSION_1 1`
- `#define SSL_MINOR_VERSION_2 2`
- `#define SSL_IS_CLIENT 0`
- `#define SSL_IS_SERVER 1`
- `#define SSL_COMPRESS_NULL 0`
- `#define SSL_VERIFY_NONE 0`
- `#define SSL_VERIFY_OPTIONAL 1`
- `#define SSL_VERIFY_REQUIRED 2`
- `#define SSL_MAX_CONTENT_LEN 16384`
- `#define SSL_BUFFER_LEN (SSL_MAX_CONTENT_LEN + 512)`
- `#define SSL_RSA_RC4_128_MD5 4`
- `#define SSL_RSA_RC4_128_SHA 5`
- `#define SSL_RSA_DES_168_SHA 10`
- `#define SSL_EDH_RSA_DES_168_SHA 22`
- `#define SSL_RSA_AES_128_SHA 47`
- `#define SSL_RSA_AES_256_SHA 53`
- `#define SSL_EDH_RSA_AES_256_SHA 57`
- `#define SSL_MSG_CHANGE_CIPHER_SPEC 20`
- `#define SSL_MSG_ALERT 21`
- `#define SSL_MSG_HANDSHAKE 22`
- `#define SSL_MSG_APPLICATION_DATA 23`
- `#define SSL_ALERT_CLOSE_NOTIFY 0`
- `#define SSL_ALERT_WARNING 1`
- `#define SSL_ALERT_FATAL 2`
- `#define SSL_ALERT_NO_CERTIFICATE 41`
- `#define SSL_HS_HELLO_REQUEST 0`
- `#define SSL_HS_CLIENT_HELLO 1`
- `#define SSL_HS_SERVER_HELLO 2`
- `#define SSL_HS_CERTIFICATE 11`
- `#define SSL_HS_SERVER_KEY_EXCHANGE 12`
- `#define SSL_HS_CERTIFICATE_REQUEST 13`
- `#define SSL_HS_SERVER_HELLO_DONE 14`
- `#define SSL_HS_CERTIFICATE_VERIFY 15`
- `#define SSL_HS_CLIENT_KEY_EXCHANGE 16`
- `#define SSL_HS_FINISHED 20`
- `#define TLS_EXT_SERVERNAME 0`
- `#define TLS_EXT_SERVERNAME_HOSTNAME 0`

## Typedefs

- `typedef struct _ssl_session ssl_session`
- `typedef struct _ssl_context ssl_context`

## Enumerations

- enum `ssl_states` {  
    `SSL_HELLO_REQUEST`, `SSL_CLIENT_HELLO`, `SSL_SERVER_HELLO`, `SSL_SERVER_CERTIFICATE`,  
    `SSL_SERVER_KEY_EXCHANGE`, `SSL_CERTIFICATE_REQUEST`, `SSL_SERVER_HELLO_DONE`, `SSL_CLIENT_CERTIFICATE`,  
    `SSL_CLIENT_KEY_EXCHANGE`, `SSL_CERTIFICATE_VERIFY`, `SSL_CLIENT_CHANGE_CIPHER_SPEC`, `SSL_CLIENT_FINISHED`,  
    `SSL_SERVER_CHANGE_CIPHER_SPEC`, `SSL_SERVER_FINISHED`, `SSL_FLUSH_BUFFERS`, `SSL_HANDSHAKE_OVER` }

## Functions

- `int ssl_init (ssl_context *ssl)`  
    Initialize an SSL context.
- `void ssl_set_endpoint (ssl_context *ssl, int endpoint)`  
    Set the current endpoint type.
- `void ssl_set_authmode (ssl_context *ssl, int authmode)`  
    Set the certificate verification mode.
- `void ssl_set_rng (ssl_context *ssl, int(*f_rng)(void *), void *p_rng)`  
    Set the random number generator callback.
- `void ssl_set_dbg (ssl_context *ssl, void(*f_dbg)(void *, int, char *), void *p_dbg)`  
    Set the debug callback.
- `void ssl_set_bio (ssl_context *ssl, int(*f_rcv)(void *, unsigned char *, int), void *p_rcv, int(*f_send)(void *, unsigned char *, int), void *p_send)`  
    Set the underlying BIO read and write callbacks.
- `void ssl_set_scb (ssl_context *ssl, int(*s_get)(ssl_context *), int(*s_set)(ssl_context *))`  
    Set the session callbacks (server-side only).
- `void ssl_set_session (ssl_context *ssl, int resume, int timeout, ssl_session *session)`  
    Set the session resuming flag, timeout and data.
- `void ssl_set_ciphers (ssl_context *ssl, int *ciphers)`  
    Set the list of allowed ciphersuites.
- `void ssl_set_ca_chain (ssl_context *ssl, x509_cert *ca_chain, char *peer_cn)`  
    Set the data required to verify peer certificate.
- `void ssl_set_own_cert (ssl_context *ssl, x509_cert *own_cert, rsa_context *rsa_key)`  
    Set own certificate and private key.
- `int ssl_set_dh_param (ssl_context *ssl, char *dhm_P, char *dhm_G)`

*Set the Diffie-Hellman public  $P$  and  $G$  values, read as hexadecimal strings (server-side only).*

- `int ssl_set_hostname (ssl_context *ssl, char *hostname)`  
*Set hostname for ServerName TLS Extension.*
- `int ssl_get_bytes_avail (ssl_context *ssl)`  
*Return the number of data bytes available to read.*
- `int ssl_get_verify_result (ssl_context *ssl)`  
*Return the result of the certificate verification.*
- `char * ssl_get_cipher (ssl_context *ssl)`  
*Return the name of the current cipher.*
- `int ssl_handshake (ssl_context *ssl)`  
*Perform the SSL handshake.*
- `int ssl_read (ssl_context *ssl, unsigned char *buf, int len)`  
*Read at most 'len' application data bytes.*
- `int ssl_write (ssl_context *ssl, unsigned char *buf, int len)`  
*Write exactly 'len' application data bytes.*
- `int ssl_close_notify (ssl_context *ssl)`  
*Notify the peer that the connection is being closed.*
- `void ssl_free (ssl_context *ssl)`  
*Free an SSL context.*
- `int ssl_handshake_client (ssl_context *ssl)`
- `int ssl_handshake_server (ssl_context *ssl)`
- `int ssl_derive_keys (ssl_context *ssl)`
- `void ssl_calc_verify (ssl_context *ssl, unsigned char hash[36])`
- `int ssl_read_record (ssl_context *ssl)`
- `int ssl_fetch_input (ssl_context *ssl, int nb_want)`
- `int ssl_write_record (ssl_context *ssl)`
- `int ssl_flush_output (ssl_context *ssl)`
- `int ssl_parse_certificate (ssl_context *ssl)`
- `int ssl_write_certificate (ssl_context *ssl)`
- `int ssl_parse_change_cipher_spec (ssl_context *ssl)`
- `int ssl_write_change_cipher_spec (ssl_context *ssl)`
- `int ssl_parse_finished (ssl_context *ssl)`
- `int ssl_write_finished (ssl_context *ssl)`

## Variables

- `int ssl_default_ciphers []`

### 13.112.1 Detailed Description

Definition in file [ssl.h](#).

### 13.112.2 Define Documentation

#### 13.112.2.1 `#define SSL_ALERT_CLOSE_NOTIFY 0`

Definition at line 87 of file [ssl.h](#).

Referenced by [ssl\\_close\\_notify\(\)](#), and [ssl\\_read\\_record\(\)](#).

#### 13.112.2.2 `#define SSL_ALERT_FATAL 2`

Definition at line 89 of file [ssl.h](#).

Referenced by [ssl\\_read\\_record\(\)](#).

#### 13.112.2.3 `#define SSL_ALERT_NO_CERTIFICATE 41`

Definition at line 90 of file [ssl.h](#).

Referenced by [ssl\\_parse\\_certificate\(\)](#), and [ssl\\_write\\_certificate\(\)](#).

#### 13.112.2.4 `#define SSL_ALERT_WARNING 1`

Definition at line 88 of file [ssl.h](#).

Referenced by [ssl\\_close\\_notify\(\)](#), [ssl\\_parse\\_certificate\(\)](#), [ssl\\_read\\_record\(\)](#), and [ssl\\_write\\_certificate\(\)](#).

#### 13.112.2.5 `#define SSL_BUFFER_LEN (SSL_MAX_CONTENT_LEN + 512)`

Definition at line 66 of file [ssl.h](#).

Referenced by [ssl\\_free\(\)](#), and [ssl\\_init\(\)](#).

#### 13.112.2.6 `#define SSL_COMPRESS_NULL 0`

Definition at line 54 of file [ssl.h](#).

Referenced by [ssl\\_parse\\_server\\_hello\(\)](#), [ssl\\_write\\_client\\_hello\(\)](#), and [ssl\\_write\\_server\\_hello\(\)](#).

#### 13.112.2.7 `#define SSL_EDH_RSA_AES_256_SHA 57`

Definition at line 77 of file [ssl.h](#).

Referenced by [main\(\)](#), [ssl\\_derive\\_keys\(\)](#), [ssl\\_get\\_cipher\(\)](#), [ssl\\_parse\\_client\\_key\\_exchange\(\)](#), [ssl\\_parse\\_server\\_key\\_exchange\(\)](#), [ssl\\_write\\_client\\_key\\_exchange\(\)](#), and [ssl\\_write\\_server\\_key\\_exchange\(\)](#).

**13.112.2.8 #define SSL\_EDH\_RSA\_DES\_168\_SHA 22**

Definition at line 74 of file ssl.h.

Referenced by main(), ssl\_derive\_keys(), ssl\_get\_cipher(), ssl\_parse\_client\_key\_exchange(), ssl\_parse\_server\_key\_exchange(), ssl\_write\_client\_key\_exchange(), and ssl\_write\_server\_key\_exchange().

**13.112.2.9 #define SSL\_HS\_CERTIFICATE 11**

Definition at line 95 of file ssl.h.

Referenced by ssl\_parse\_certificate(), and ssl\_write\_certificate().

**13.112.2.10 #define SSL\_HS\_CERTIFICATE\_REQUEST 13**

Definition at line 97 of file ssl.h.

Referenced by ssl\_parse\_certificate\_request(), and ssl\_write\_certificate\_request().

**13.112.2.11 #define SSL\_HS\_CERTIFICATE\_VERIFY 15**

Definition at line 99 of file ssl.h.

Referenced by ssl\_parse\_certificate\_verify(), and ssl\_write\_certificate\_verify().

**13.112.2.12 #define SSL\_HS\_CLIENT\_HELLO 1**

Definition at line 93 of file ssl.h.

Referenced by ssl\_parse\_client\_hello(), and ssl\_write\_client\_hello().

**13.112.2.13 #define SSL\_HS\_CLIENT\_KEY\_EXCHANGE 16**

Definition at line 100 of file ssl.h.

Referenced by ssl\_parse\_client\_key\_exchange(), and ssl\_write\_client\_key\_exchange().

**13.112.2.14 #define SSL\_HS\_FINISHED 20**

Definition at line 101 of file ssl.h.

Referenced by ssl\_parse\_finished(), and ssl\_write\_finished().

**13.112.2.15 #define SSL\_HS\_HELLO\_REQUEST 0**

Definition at line 92 of file ssl.h.

**13.112.2.16 #define SSL\_HS\_SERVER\_HELLO 2**

Definition at line 94 of file ssl.h.

Referenced by ssl\_parse\_server\_hello(), and ssl\_write\_server\_hello().

**13.112.2.17 #define SSL\_HS\_SERVER\_HELLO\_DONE 14**

Definition at line 98 of file ssl.h.

Referenced by ssl\_parse\_server\_hello\_done(), and ssl\_write\_server\_hello\_done().

**13.112.2.18 #define SSL\_HS\_SERVER\_KEY\_EXCHANGE 12**

Definition at line 96 of file ssl.h.

Referenced by ssl\_parse\_server\_key\_exchange(), and ssl\_write\_server\_key\_exchange().

**13.112.2.19 #define SSL\_IS\_CLIENT 0**

Definition at line 52 of file ssl.h.

Referenced by main(), ssl\_calc\_finished(), ssl\_derive\_keys(), ssl\_handshake(), ssl\_parse\_finished(), ssl\_test(), ssl\_write\_certificate(), and ssl\_write\_finished().

**13.112.2.20 #define SSL\_IS\_SERVER 1**

Definition at line 53 of file ssl.h.

Referenced by main(), ssl\_handshake(), ssl\_parse\_certificate(), ssl\_parse\_finished(), and ssl\_test().

**13.112.2.21 #define SSL\_MAJOR\_VERSION\_3 3**

Definition at line 47 of file ssl.h.

Referenced by ssl\_parse\_client\_hello(), ssl\_parse\_server\_hello(), and ssl\_write\_client\_hello().

**13.112.2.22 #define SSL\_MAX\_CONTENT\_LEN 16384**

Definition at line 60 of file ssl.h.

Referenced by ssl\_read\_record(), ssl\_write(), and ssl\_write\_certificate().

**13.112.2.23 #define SSL\_MINOR\_VERSION\_0 0**

SSL v3.0

Definition at line 48 of file ssl.h.

Referenced by ssl\_calc\_finished(), ssl\_calc\_verify(), ssl\_decrypt\_buf(), ssl\_derive\_keys(), ssl\_encrypt\_buf(), ssl\_parse\_certificate(), ssl\_parse\_client\_key\_exchange(), ssl\_parse\_finished(), ssl\_parse\_server\_hello(), ssl\_read\_record(), ssl\_write\_certificate(), ssl\_write\_client\_hello(), ssl\_write\_client\_key\_exchange(), and ssl\_write\_finished().

**13.112.2.24 #define SSL\_MINOR\_VERSION\_1 1**

TLS v1.0

Definition at line 49 of file ssl.h.

Referenced by `ssl_parse_client_hello()`, `ssl_parse_server_hello()`, `ssl_read_record()`, and `ssl_write_client_hello()`.

#### **13.112.2.25 #define SSL\_MINOR\_VERSION\_2 2**

TLS v1.1

Definition at line 50 of file `ssl.h`.

#### **13.112.2.26 #define SSL\_MSG\_ALERT 21**

Definition at line 83 of file `ssl.h`.

Referenced by `ssl_close_notify()`, `ssl_parse_certificate()`, `ssl_read_record()`, and `ssl_write_certificate()`.

#### **13.112.2.27 #define SSL\_MSG\_APPLICATION\_DATA 23**

Definition at line 85 of file `ssl.h`.

Referenced by `ssl_read()`, and `ssl_write()`.

#### **13.112.2.28 #define SSL\_MSG\_CHANGE\_CIPHER\_SPEC 20**

Definition at line 82 of file `ssl.h`.

Referenced by `ssl_parse_change_cipher_spec()`, and `ssl_write_change_cipher_spec()`.

#### **13.112.2.29 #define SSL\_MSG\_HANDSHAKE 22**

Definition at line 84 of file `ssl.h`.

Referenced by `ssl_parse_certificate()`, `ssl_parse_certificate_request()`, `ssl_parse_certificate_verify()`, `ssl_parse_client_hello()`, `ssl_parse_client_key_exchange()`, `ssl_parse_finished()`, `ssl_parse_server_hello()`, `ssl_parse_server_hello_done()`, `ssl_parse_server_key_exchange()`, `ssl_read_record()`, `ssl_write_certificate()`, `ssl_write_certificate_request()`, `ssl_write_certificate_verify()`, `ssl_write_client_hello()`, `ssl_write_client_key_exchange()`, `ssl_write_finished()`, `ssl_write_record()`, `ssl_write_server_hello()`, `ssl_write_server_hello_done()`, and `ssl_write_server_key_exchange()`.

#### **13.112.2.30 #define SSL\_RSA\_AES\_128\_SHA 47**

Definition at line 75 of file `ssl.h`.

Referenced by `main()`, `ssl_derive_keys()`, and `ssl_get_cipher()`.

#### **13.112.2.31 #define SSL\_RSA\_AES\_256\_SHA 53**

Definition at line 76 of file `ssl.h`.

Referenced by `main()`, `ssl_derive_keys()`, and `ssl_get_cipher()`.



**13.112.2.32 #define SSL\_RSA\_DES\_168\_SHA 10**

Definition at line 73 of file ssl.h.

Referenced by main(), ssl\_derive\_keys(), and ssl\_get\_cipher().

**13.112.2.33 #define SSL\_RSA\_RC4\_128\_MD5 4**

Definition at line 71 of file ssl.h.

Referenced by main(), ssl\_derive\_keys(), and ssl\_get\_cipher().

**13.112.2.34 #define SSL\_RSA\_RC4\_128\_SHA 5**

Definition at line 72 of file ssl.h.

Referenced by main(), ssl\_derive\_keys(), and ssl\_get\_cipher().

**13.112.2.35 #define SSL\_VERIFY\_NONE 0**

Definition at line 56 of file ssl.h.

Referenced by main(), ssl\_parse\_certificate(), ssl\_test(), and ssl\_write\_certificate\_request().

**13.112.2.36 #define SSL\_VERIFY\_OPTIONAL 1**

Definition at line 57 of file ssl.h.

Referenced by main(), and ssl\_parse\_certificate().

**13.112.2.37 #define SSL\_VERIFY\_REQUIRED 2**

Definition at line 58 of file ssl.h.

Referenced by ssl\_parse\_certificate().

**13.112.2.38 #define TLS\_EXT\_SERVERNAME 0**

Definition at line 106 of file ssl.h.

Referenced by ssl\_write\_client\_hello().

**13.112.2.39 #define TLS\_EXT\_SERVERNAME\_HOSTNAME 0**

Definition at line 107 of file ssl.h.

Referenced by ssl\_write\_client\_hello().

**13.112.2.40 #define XYSSL\_ERR\_SSL\_BAD\_HS\_CERTIFICATE -0xA800**

Definition at line 35 of file ssl.h.

Referenced by ssl\_parse\_certificate().

**13.112.2.41 #define XYSSL\_ERR\_SSL\_BAD\_HS\_CERTIFICATE\_REQUEST -0xB000**

Definition at line 36 of file ssl.h.

**13.112.2.42 #define XYSSL\_ERR\_SSL\_BAD\_HS\_CERTIFICATE\_VERIFY -0xD000**

Definition at line 40 of file ssl.h.

Referenced by ssl\_parse\_certificate\_verify().

**13.112.2.43 #define XYSSL\_ERR\_SSL\_BAD\_HS\_CHANGE\_CIPHER\_SPEC -0xD800**

Definition at line 41 of file ssl.h.

Referenced by ssl\_parse\_change\_cipher\_spec().

**13.112.2.44 #define XYSSL\_ERR\_SSL\_BAD\_HS\_CLIENT\_HELLO -0x9800**

Definition at line 33 of file ssl.h.

Referenced by ssl\_parse\_client\_hello().

**13.112.2.45 #define XYSSL\_ERR\_SSL\_BAD\_HS\_CLIENT\_KEY\_EXCHANGE -0xC800**

Definition at line 39 of file ssl.h.

Referenced by ssl\_parse\_client\_key\_exchange().

**13.112.2.46 #define XYSSL\_ERR\_SSL\_BAD\_HS\_FINISHED -0xE000**

Definition at line 42 of file ssl.h.

Referenced by ssl\_parse\_finished().

**13.112.2.47 #define XYSSL\_ERR\_SSL\_BAD\_HS\_SERVER\_HELLO -0xA000**

Definition at line 34 of file ssl.h.

Referenced by ssl\_parse\_server\_hello().

**13.112.2.48 #define XYSSL\_ERR\_SSL\_BAD\_HS\_SERVER\_HELLO\_DONE -0xC000**

Definition at line 38 of file ssl.h.

Referenced by ssl\_parse\_server\_hello\_done().

**13.112.2.49 #define XYSSL\_ERR\_SSL\_BAD\_HS\_SERVER\_KEY\_EXCHANGE -0xB800**

Definition at line 37 of file ssl.h.

Referenced by ssl\_parse\_server\_key\_exchange().

**13.112.2.50 #define XYSSL\_ERR\_SSL\_BAD\_INPUT\_DATA -0x1800**

Definition at line 17 of file ssl.h.

Referenced by ssl\_handshake\_client(), ssl\_handshake\_server(), ssl\_set\_hostname(), and tls1\_prf().

**13.112.2.51 #define XYSSL\_ERR\_SSL\_CA\_CHAIN\_REQUIRED -0x7000**

Definition at line 28 of file ssl.h.

Referenced by ssl\_parse\_certificate().

**13.112.2.52 #define XYSSL\_ERR\_SSL\_CERTIFICATE\_REQUIRED -0x6000**

Definition at line 26 of file ssl.h.

Referenced by ssl\_write\_certificate().

**13.112.2.53 #define XYSSL\_ERR\_SSL\_CERTIFICATE\_TOO\_LARGE -0x5800**

Definition at line 25 of file ssl.h.

Referenced by ssl\_write\_certificate().

**13.112.2.54 #define XYSSL\_ERR\_SSL\_FATAL\_ALERT\_MESSAGE -0x8000**

Definition at line 30 of file ssl.h.

Referenced by ssl\_read\_record().

**13.112.2.55 #define XYSSL\_ERR\_SSL\_FEATURE\_UNAVAILABLE -0x1000**

Definition at line 16 of file ssl.h.

Referenced by ssl\_decrypt\_buf(), ssl\_derive\_keys(), ssl\_encrypt\_buf(), ssl\_handshake(), ssl\_parse\_client\_key\_exchange(), ssl\_parse\_server\_key\_exchange(), ssl\_write\_client\_key\_exchange(), and ssl\_write\_server\_key\_exchange().

**13.112.2.56 #define XYSSL\_ERR\_SSL\_INVALID\_MAC -0x2000**

Definition at line 18 of file ssl.h.

Referenced by ssl\_decrypt\_buf().

**13.112.2.57 #define XYSSL\_ERR\_SSL\_INVALID\_MODULUS\_SIZE -0x3000**

Definition at line 20 of file ssl.h.

**13.112.2.58 #define XYSSL\_ERR\_SSL\_INVALID\_RECORD -0x2800**

Definition at line 19 of file ssl.h.

Referenced by ssl\_read\_record().

**13.112.2.59 #define XYSSL\_ERR\_SSL\_NO\_CIPHER\_CHOSEN -0x4000**

Definition at line 22 of file ssl.h.

Referenced by ssl\_parse\_client\_hello().

**13.112.2.60 #define XYSSL\_ERR\_SSL\_NO\_CLIENT\_CERTIFICATE -0x5000**

Definition at line 24 of file ssl.h.

Referenced by ssl\_parse\_certificate().

**13.112.2.61 #define XYSSL\_ERR\_SSL\_NO\_SESSION\_FOUND -0x4800**

Definition at line 23 of file ssl.h.

**13.112.2.62 #define XYSSL\_ERR\_SSL\_PEER\_CLOSE\_NOTIFY -0x9000**

Definition at line 32 of file ssl.h.

Referenced by main(), ssl\_read\_record(), and ssl\_test().

**13.112.2.63 #define XYSSL\_ERR\_SSL\_PEER\_VERIFY\_FAILED -0x8800**

Definition at line 31 of file ssl.h.

**13.112.2.64 #define XYSSL\_ERR\_SSL\_PRIVATE\_KEY\_REQUIRED -0x6800**

Definition at line 27 of file ssl.h.

Referenced by ssl\_write\_certificate\_verify().

**13.112.2.65 #define XYSSL\_ERR\_SSL\_UNEXPECTED\_MESSAGE -0x7800**

Definition at line 29 of file ssl.h.

Referenced by ssl\_parse\_certificate(), ssl\_parse\_certificate\_request(), ssl\_parse\_change\_cipher\_spec(), ssl\_parse\_finished(), ssl\_parse\_server\_hello(), ssl\_parse\_server\_hello\_done(), ssl\_parse\_server\_key\_exchange(), and ssl\_read().

**13.112.2.66 #define XYSSL\_ERR\_SSL\_UNKNOWN\_CIPHER -0x3800**

Definition at line 21 of file ssl.h.

### 13.112.3 Typedef Documentation

#### 13.112.3.1 typedef struct \_ssl\_context ssl\_context

Definition at line 134 of file ssl.h.

#### 13.112.3.2 typedef struct \_ssl\_session ssl\_session

Definition at line 133 of file ssl.h.

### 13.112.4 Enumeration Type Documentation

#### 13.112.4.1 enum ssl\_states

Enumerator:

*SSL\_HELLO\_REQUEST*  
*SSL\_CLIENT\_HELLO*  
*SSL\_SERVER\_HELLO*  
*SSL\_SERVER\_CERTIFICATE*  
*SSL\_SERVER\_KEY\_EXCHANGE*  
*SSL\_CERTIFICATE\_REQUEST*  
*SSL\_SERVER\_HELLO\_DONE*  
*SSL\_CLIENT\_CERTIFICATE*  
*SSL\_CLIENT\_KEY\_EXCHANGE*  
*SSL\_CERTIFICATE\_VERIFY*  
*SSL\_CLIENT\_CHANGE\_CIPHER\_SPEC*  
*SSL\_CLIENT\_FINISHED*  
*SSL\_SERVER\_CHANGE\_CIPHER\_SPEC*  
*SSL\_SERVER\_FINISHED*  
*SSL\_FLUSH\_BUFFERS*  
*SSL\_HANDSHAKE\_OVER*

Definition at line 112 of file ssl.h.

### 13.112.5 Function Documentation

#### 13.112.5.1 void ssl\_calc\_verify (ssl\_context \* *ssl*, unsigned char *hash*[36])

Definition at line 335 of file ssl\_tls.c.

References `_ssl_context::fin_md5`, `_ssl_context::fin_sha1`, `_ssl_session::master`, `md5()`, `md5_finish()`, `md5_starts()`, `md5_update()`, `_ssl_context::minor_ver`, `_ssl_context::session`, `sha1()`, `sha1_finish()`, `sha1_starts()`, `sha1_update()`, `SSL_DEBUG_BUF`, `SSL_DEBUG_MSG`, and `SSL_MINOR_VERSION_0`.

Referenced by `ssl_parse_certificate_verify()`, and `ssl_write_certificate_verify()`.

**13.112.5.2 int ssl\_close\_notify (ssl\_context \* ssl)**

Notify the peer that the connection is being closed.

Definition at line 1904 of file ssl\_tls.c.

References `_ssl_context::out_msg`, `_ssl_context::out_msglen`, `_ssl_context::out_msgtype`, `SSL_ALERT_CLOSE_NOTIFY`, `SSL_ALERT_WARNING`, `SSL_DEBUG_MSG`, `SSL_DEBUG_RET`, `ssl_flush_output()`, `SSL_HANDSHAKE_OVER`, `SSL_MSG_ALERT`, `ssl_write_record()`, and `_ssl_context::state`.

Referenced by `main()`, and `ssl_test()`.

**13.112.5.3 int ssl\_derive\_keys (ssl\_context \* ssl)**

Definition at line 106 of file ssl\_tls.c.

References `aes_setkey_dec()`, `aes_setkey_enc()`, `arc4_setup()`, `_ssl_session::cipher`, `_ssl_context::ctx_dec`, `_ssl_context::ctx_enc`, `des3_set3key_dec()`, `des3_set3key_enc()`, `_ssl_context::endpoint`, `_ssl_context::iv_dec`, `_ssl_context::iv_enc`, `_ssl_context::ivlen`, `_ssl_context::keylen`, `_ssl_context::mac_dec`, `_ssl_context::mac_enc`, `_ssl_context::maclen`, `_ssl_session::master`, `md5()`, `md5_finish()`, `md5_starts()`, `md5_update()`, `_ssl_context::minlen`, `_ssl_context::minor_ver`, `_ssl_context::pmslen`, `_ssl_context::premaster`, `_ssl_context::randbytes`, `_ssl_context::resume`, `_ssl_context::session`, `sha1()`, `sha1_finish()`, `sha1_starts()`, `sha1_update()`, `SSL_DEBUG_BUF`, `SSL_DEBUG_MSG`, `SSL_EDH_RSA_AES_256_SHA`, `SSL_EDH_RSA_DES_168_SHA`, `ssl_get_cipher()`, `SSL_IS_CLIENT`, `SSL_MINOR_VERSION_0`, `SSL_RSA_AES_128_SHA`, `SSL_RSA_AES_256_SHA`, `SSL_RSA_DES_168_SHA`, `SSL_RSA_RC4_128_MD5`, `SSL_RSA_RC4_128_SHA`, `tls1_prf()`, and `XYSSL_ERR_SSL_FEATURE_UNAVAILABLE`.

Referenced by `ssl_parse_client_key_exchange()`, `ssl_parse_server_hello()`, `ssl_write_client_key_exchange()`, and `ssl_write_server_hello()`.

**13.112.5.4 int ssl\_fetch\_input (ssl\_context \* ssl, int nb\_want)**

Definition at line 727 of file ssl\_tls.c.

References `_ssl_context::f_recv`, `_ssl_context::in_hdr`, `_ssl_context::in_left`, `_ssl_context::p_recv`, `SSL_DEBUG_MSG`, and `SSL_DEBUG_RET`.

Referenced by `ssl_parse_client_hello()`, and `ssl_read_record()`.

**13.112.5.5 int ssl\_flush\_output (ssl\_context \* ssl)**

Definition at line 756 of file ssl\_tls.c.

References `buf`, `_ssl_context::f_send`, `_ssl_context::out_hdr`, `_ssl_context::out_left`, `_ssl_context::out_msglen`, `_ssl_context::p_send`, `SSL_DEBUG_MSG`, and `SSL_DEBUG_RET`.

Referenced by `ssl_close_notify()`, `ssl_handshake_client()`, `ssl_handshake_server()`, `ssl_write()`, and `ssl_write_record()`.

**13.112.5.6 void ssl\_free (ssl\_context \* ssl)**

Free an SSL context.

Definition at line 1938 of file ssl\_tls.c.

References `_ssl_context::dhm_ctx`, `dhm_free()`, `_ssl_context::hostname`, `_ssl_context::hostname_len`, `_ssl_context::in_ctr`, `_ssl_context::out_ctr`, `_ssl_context::peer_cert`, `SSL_BUFFER_LEN`, `SSL_DEBUG_MSG`, and `x509_free()`.

Referenced by `main()`, and `ssl_test()`.

#### 13.112.5.7 `int ssl_get_bytes_avail (ssl_context * ssl)`

Return the number of data bytes available to read.

##### Parameters:

*ssl* SSL context

##### Returns:

how many bytes are available in the read buffer

Definition at line 1691 of file `ssl_tls.c`.

References `_ssl_context::in_msglen`, and `_ssl_context::in_offt`.

#### 13.112.5.8 `char* ssl_get_cipher (ssl_context * ssl)`

Return the name of the current cipher.

##### Parameters:

*ssl* SSL context

##### Returns:

a string containing the cipher name

Definition at line 1701 of file `ssl_tls.c`.

References `_ssl_session::cipher`, `_ssl_context::session`, `SSL_EDH_RSA_AES_256_SHA`, `SSL_EDH_RSA_DES_168_SHA`, `SSL_RSA_AES_128_SHA`, `SSL_RSA_AES_256_SHA`, `SSL_RSA_DES_168_SHA`, `SSL_RSA_RC4_128_MD5`, and `SSL_RSA_RC4_128_SHA`.

Referenced by `main()`, and `ssl_derive_keys()`.

#### 13.112.5.9 `int ssl_get_verify_result (ssl_context * ssl)`

Return the result of the certificate verification.

##### Parameters:

*ssl* SSL context

##### Returns:

0 if successful, or a combination of: `BADCERT_EXPIRED` `BADCERT_REVOKED` `BADCERT_CN_MISMATCH` `BADCERT_NOT_TRUSTED`

Definition at line 1696 of file `ssl_tls.c`.

References `_ssl_context::verify_result`.

Referenced by `main()`.

**13.112.5.10 int ssl\_handshake (ssl\_context \* ssl)**

Perform the SSL handshake.

**Parameters:**

*ssl* SSL context

**Returns:**

0 if successful, XYSSL\_ERR\_NET\_TRY\_AGAIN, or a specific SSL error code.

Definition at line 1767 of file ssl\_tls.c.

References `_ssl_context::endpoint`, `SSL_DEBUG_MSG`, `ssl_handshake_client()`, `ssl_handshake_server()`, `SSL_IS_CLIENT`, `SSL_IS_SERVER`, and `XYSSL_ERR_SSL_FEATURE_UNAVAILABLE`.

Referenced by `main()`, `ssl_read()`, and `ssl_write()`.

**13.112.5.11 int ssl\_handshake\_client (ssl\_context \* ssl)**

Definition at line 660 of file ssl\_cli.c.

References `SSL_CERTIFICATE_REQUEST`, `SSL_CERTIFICATE_VERIFY`, `SSL_CLIENT_CERTIFICATE`, `SSL_CLIENT_CHANGE_CIPHER_SPEC`, `SSL_CLIENT_FINISHED`, `SSL_CLIENT_HELLO`, `SSL_CLIENT_KEY_EXCHANGE`, `SSL_DEBUG_MSG`, `SSL_FLUSH_BUFFERS`, `ssl_flush_output()`, `SSL_HANDSHAKE_OVER`, `SSL_HELLO_REQUEST`, `ssl_parse_certificate()`, `ssl_parse_certificate_request()`, `ssl_parse_change_cipher_spec()`, `ssl_parse_finished()`, `ssl_parse_server_hello()`, `ssl_parse_server_hello_done()`, `ssl_parse_server_key_exchange()`, `SSL_SERVER_CERTIFICATE`, `SSL_SERVER_CHANGE_CIPHER_SPEC`, `SSL_SERVER_FINISHED`, `SSL_SERVER_HELLO`, `SSL_SERVER_HELLO_DONE`, `SSL_SERVER_KEY_EXCHANGE`, `ssl_write_certificate()`, `ssl_write_certificate_verify()`, `ssl_write_change_cipher_spec()`, `ssl_write_client_hello()`, `ssl_write_client_key_exchange()`, `ssl_write_finished()`, `_ssl_context::state`, and `XYSSL_ERR_SSL_BAD_INPUT_DATA`.

Referenced by `ssl_handshake()`.

**13.112.5.12 int ssl\_handshake\_server (ssl\_context \* ssl)**

Definition at line 819 of file ssl\_srv.c.

References `SSL_CERTIFICATE_REQUEST`, `SSL_CERTIFICATE_VERIFY`, `SSL_CLIENT_CERTIFICATE`, `SSL_CLIENT_CHANGE_CIPHER_SPEC`, `SSL_CLIENT_FINISHED`, `SSL_CLIENT_HELLO`, `SSL_CLIENT_KEY_EXCHANGE`, `SSL_DEBUG_MSG`, `SSL_FLUSH_BUFFERS`, `ssl_flush_output()`, `SSL_HANDSHAKE_OVER`, `SSL_HELLO_REQUEST`, `ssl_parse_certificate()`, `ssl_parse_certificate_verify()`, `ssl_parse_change_cipher_spec()`, `ssl_parse_client_hello()`, `ssl_parse_client_key_exchange()`, `ssl_parse_finished()`, `SSL_SERVER_CERTIFICATE`, `SSL_SERVER_CHANGE_CIPHER_SPEC`, `SSL_SERVER_FINISHED`, `SSL_SERVER_HELLO`, `SSL_SERVER_HELLO_DONE`, `SSL_SERVER_KEY_EXCHANGE`, `ssl_write_certificate()`, `ssl_write_certificate_request()`, `ssl_write_change_cipher_spec()`, `ssl_write_finished()`, `ssl_write_server_hello()`, `ssl_write_server_hello_done()`, `ssl_write_server_key_exchange()`, `_ssl_context::state`, and `XYSSL_ERR_SSL_BAD_INPUT_DATA`.

Referenced by `ssl_handshake()`.

**13.112.5.13 int ssl\_init (ssl\_context \* ssl)**

Initialize an SSL context.



**Parameters:**

*ssl* SSL context

**Returns:**

0 if successful, or 1 if memory allocation failed

Definition at line 1542 of file ssl\_tls.c.

References `_ssl_context::fin_md5`, `_ssl_context::fin_sha1`, `_ssl_context::hostname`, `_ssl_context::hostname_len`, `_ssl_context::in_ctr`, `_ssl_context::in_hdr`, `_ssl_context::in_msg`, `md5_starts()`, `_ssl_context::out_ctr`, `_ssl_context::out_hdr`, `_ssl_context::out_msg`, `sha1_starts()`, `SSL_BUFFER_LEN`, and `SSL_DEBUG_MSG`.

Referenced by `main()`, and `ssl_test()`.

**13.112.5.14 `int ssl_parse_certificate(ssl_context * ssl)`**

Definition at line 1140 of file ssl\_tls.c.

References `_ssl_context::authmode`, `_ssl_context::ca_chain`, `_ssl_context::endpoint`, `_ssl_context::in_hhlen`, `_ssl_context::in_msg`, `_ssl_context::in_msglen`, `_ssl_context::in_msgtype`, `int`, `_ssl_context::minor_ver`, `_ssl_context::peer_cert`, `_ssl_context::peer_cn`, `SSL_ALERT_NO_CERTIFICATE`, `SSL_ALERT_WARNING`, `SSL_DEBUG_CRT`, `SSL_DEBUG_MSG`, `SSL_DEBUG_RET`, `SSL_HS_CERTIFICATE`, `SSL_IS_SERVER`, `SSL_MINOR_VERSION_0`, `SSL_MSG_ALERT`, `SSL_MSG_HANDSHAKE`, `ssl_read_record()`, `SSL_VERIFY_NONE`, `SSL_VERIFY_OPTIONAL`, `SSL_VERIFY_REQUIRED`, `_ssl_context::state`, `_ssl_context::verify_result`, `x509parse_cert()`, `x509parse_verify()`, `XYSSL_ERR_SSL_BAD_HS_CERTIFICATE`, `XYSSL_ERR_SSL_CA_CHAIN_REQUIRED`, `XYSSL_ERR_SSL_NO_CLIENT_CERTIFICATE`, and `XYSSL_ERR_SSL_UNEXPECTED_MESSAGE`.

Referenced by `ssl_handshake_client()`, and `ssl_handshake_server()`.

**13.112.5.15 `int ssl_parse_change_cipher_spec(ssl_context * ssl)`**

Definition at line 1311 of file ssl\_tls.c.

References `_ssl_context::do_crypt`, `_ssl_context::in_msg`, `_ssl_context::in_msglen`, `_ssl_context::in_msgtype`, `SSL_DEBUG_MSG`, `SSL_DEBUG_RET`, `SSL_MSG_CHANGE_CIPHER_SPEC`, `ssl_read_record()`, `_ssl_context::state`, `XYSSL_ERR_SSL_BAD_HS_CHANGE_CIPHER_SPEC`, and `XYSSL_ERR_SSL_UNEXPECTED_MESSAGE`.

Referenced by `ssl_handshake_client()`, and `ssl_handshake_server()`.

**13.112.5.16 `int ssl_parse_finished(ssl_context * ssl)`**

Definition at line 1480 of file ssl\_tls.c.

References `buf`, `_ssl_context::do_crypt`, `_ssl_context::endpoint`, `_ssl_context::fin_md5`, `_ssl_context::fin_sha1`, `_ssl_context::in_hhlen`, `_ssl_context::in_msg`, `_ssl_context::in_msgtype`, `md5()`, `_ssl_context::minor_ver`, `_ssl_context::resume`, `sha1()`, `ssl_calc_finished()`, `SSL_CLIENT_CHANGE_CIPHER_SPEC`, `SSL_DEBUG_MSG`, `SSL_DEBUG_RET`, `SSL_HANDSHAKE_OVER`, `SSL_HS_FINISHED`, `SSL_IS_CLIENT`, `SSL_IS_SERVER`, `SSL_MINOR_VERSION_0`, `SSL_MSG_HANDSHAKE`, `ssl_read_record()`, `_ssl_context::state`, `XYSSL_ERR_SSL_BAD_HS_FINISHED`, and `XYSSL_ERR_SSL_UNEXPECTED_MESSAGE`.

Referenced by `ssl_handshake_client()`, and `ssl_handshake_server()`.

**13.112.5.17 int ssl\_read (ssl\_context \* ssl, unsigned char \* buf, int len)**

Read at most 'len' application data bytes.

**Parameters:**

*ssl* SSL context  
*buf* buffer that will hold the data  
*len* how many bytes must be read

**Returns:**

This function returns the number of bytes read, or a negative error code.

Definition at line 1791 of file ssl\_tls.c.

References `_ssl_context::in_msg`, `_ssl_context::in_msglen`, `_ssl_context::in_msgtype`, `_ssl_context::in_offt`, `SSL_DEBUG_MSG`, `SSL_DEBUG_RET`, `ssl_handshake()`, `SSL_HANDSHAKE_OVER`, `SSL_MSG_APPLICATION_DATA`, `ssl_read_record()`, `_ssl_context::state`, and `XYSSL_ERR_SSL_UNEXPECTED_MESSAGE`.

Referenced by `main()`, and `ssl_test()`.

**13.112.5.18 int ssl\_read\_record (ssl\_context \* ssl)**

Definition at line 842 of file ssl\_tls.c.

References `_ssl_context::do_crypt`, `_ssl_context::fin_md5`, `_ssl_context::fin_sha1`, `_ssl_context::in_hdr`, `_ssl_context::in_hlen`, `_ssl_context::in_left`, `_ssl_context::in_msg`, `_ssl_context::in_msglen`, `_ssl_context::in_msgtype`, `_ssl_context::major_ver`, `md5_update()`, `_ssl_context::minlen`, `_ssl_context::minor_ver`, `sha1_update()`, `SSL_ALERT_CLOSE_NOTIFY`, `SSL_ALERT_FATAL`, `SSL_ALERT_WARNING`, `SSL_DEBUG_BUF`, `SSL_DEBUG_MSG`, `SSL_DEBUG_RET`, `ssl_decrypt_buf()`, `ssl_fetch_input()`, `SSL_MAX_CONTENT_LEN`, `SSL_MINOR_VERSION_0`, `SSL_MINOR_VERSION_1`, `SSL_MSG_ALERT`, `SSL_MSG_HANDSHAKE`, `XYSSL_ERR_SSL_FATAL_ALERT_MESSAGE`, `XYSSL_ERR_SSL_INVALID_RECORD`, and `XYSSL_ERR_SSL_PEER_CLOSE_NOTIFY`.

Referenced by `ssl_parse_certificate()`, `ssl_parse_certificate_request()`, `ssl_parse_certificate_verify()`, `ssl_parse_change_cipher_spec()`, `ssl_parse_client_key_exchange()`, `ssl_parse_finished()`, `ssl_parse_server_hello()`, `ssl_parse_server_hello_done()`, `ssl_parse_server_key_exchange()`, and `ssl_read()`.

**13.112.5.19 void ssl\_set\_authmode (ssl\_context \* ssl, int authmode)**

Set the certificate verification mode.

**Parameters:**

*ssl* SSL context  
*mode* can be:

`SSL_VERIFY_NONE`: peer certificate is not checked (default), this is insecure and SHOULD be avoided.

`SSL_VERIFY_OPTIONAL`: peer certificate is checked, however the handshake continues even if verification failed; `ssl_get_verify_result()` can be called after the handshake is complete.

`SSL_VERIFY_REQUIRED`: peer *must* present a valid certificate, handshake is aborted if verification failed.

Definition at line 1589 of file ssl\_tls.c.

References `_ssl_context::authmode`.

Referenced by `main()`, and `ssl_test()`.

**13.112.5.20** `void ssl_set_bio (ssl_context * ssl, int(*) (void *, unsigned char *, int) f_recv, void * p_recv, int(*) (void *, unsigned char *, int) f_send, void * p_send)`

Set the underlying BIO read and write callbacks.

**Parameters:**

*ssl* SSL context

*f\_recv* read callback

*p\_recv* read parameter

*f\_send* write callback

*p\_send* write parameter

Definition at line 1610 of file ssl\_tls.c.

References `_ssl_context::f_recv`, `_ssl_context::f_send`, `_ssl_context::p_recv`, and `_ssl_context::p_send`.

Referenced by `main()`, and `ssl_test()`.

**13.112.5.21** `void ssl_set_ca_chain (ssl_context * ssl, x509_cert * ca_chain, char * peer_cn)`

Set the data required to verify peer certificate.

**Parameters:**

*ssl* SSL context

*ca\_chain* trusted CA chain

*peer\_cn* expected peer CommonName (or NULL)

**Note:**

TODO: add two more parameters: depth and crl

Definition at line 1641 of file ssl\_tls.c.

References `_ssl_context::ca_chain`, and `_ssl_context::peer_cn`.

Referenced by `main()`, and `ssl_test()`.

**13.112.5.22** `void ssl_set_ciphers (ssl_context * ssl, int * ciphers)`

Set the list of allowed ciphersuites.

**Parameters:**

*ssl* SSL context

*ciphers* 0-terminated list of allowed ciphers

Definition at line 1636 of file `ssl_tls.c`.

References `_ssl_context::ciphers`.

Referenced by `main()`, and `ssl_test()`.

### 13.112.5.23 `void ssl_set_dbg (ssl_context * ssl, void(*)(void *, int, char *) f_dbg, void * p_dbg)`

Set the debug callback.

#### Parameters:

*ssl* SSL context

*f\_dbg* debug function

*p\_dbg* debug parameter

Definition at line 1602 of file `ssl_tls.c`.

References `_ssl_context::f_dbg`, and `_ssl_context::p_dbg`.

Referenced by `main()`, and `ssl_test()`.

### 13.112.5.24 `int ssl_set_dh_param (ssl_context * ssl, char * dhm_P, char * dhm_G)`

Set the Diffie-Hellman public P and G values, read as hexadecimal strings (server-side only).

#### Parameters:

*ssl* SSL context

*dhm\_P* Diffie-Hellman-Merkle modulus

*dhm\_G* Diffie-Hellman-Merkle generator

#### Returns:

0 if successful

Definition at line 1655 of file `ssl_tls.c`.

References `_ssl_context::dhm_ctx`, `dhm_context::G`, `mpi_read_string()`, `dhm_context::P`, and `SSL_DEBUG_RET`.

Referenced by `main()`, and `ssl_test()`.

### 13.112.5.25 `void ssl_set_endpoint (ssl_context * ssl, int endpoint)`

Set the current endpoint type.

#### Parameters:

*ssl* SSL context

*endpoint* must be `SSL_IS_CLIENT` or `SSL_IS_SERVER`

Definition at line 1584 of file `ssl_tls.c`.

References `_ssl_context::endpoint`.

Referenced by `main()`, and `ssl_test()`.

#### 13.112.5.26 `int ssl_set_hostname (ssl_context * ssl, char * hostname)`

Set hostname for ServerName TLS Extension.

##### Parameters:

*ssl* SSL context  
*hostname* the server hostname

##### Returns:

0 if successful

Definition at line 1674 of file ssl\_tls.c.

References `_ssl_context::hostname`, `_ssl_context::hostname_len`, and `XYSSL_ERR_SSL_BAD_INPUT_DATA`.

Referenced by `main()`.

#### 13.112.5.27 `void ssl_set_own_cert (ssl_context * ssl, x509_cert * own_cert, rsa_context * rsa_key)`

Set own certificate and private key.

##### Parameters:

*ssl* SSL context  
*own\_cert* own public certificate  
*rsa\_key* own private RSA key

Definition at line 1648 of file ssl\_tls.c.

References `_ssl_context::own_cert`, and `_ssl_context::rsa_key`.

Referenced by `main()`, and `ssl_test()`.

#### 13.112.5.28 `void ssl_set_rng (ssl_context * ssl, int(*) (void *) f_rng, void * p_rng)`

Set the random number generator callback.

##### Parameters:

*ssl* SSL context  
*f\_rng* RNG function  
*p\_rng* RNG parameter

Definition at line 1594 of file ssl\_tls.c.

References `_ssl_context::f_rng`, and `_ssl_context::p_rng`.

Referenced by `main()`, and `ssl_test()`.

#### 13.112.5.29 `void ssl_set_scb (ssl_context * ssl, int(*) (ssl_context *) s_get, int(*) (ssl_context *) s_set)`

Set the session callbacks (server-side only).

**Parameters:**

*ssl* SSL context  
*s\_get* session get callback  
*s\_set* session set callback

Definition at line 1620 of file `ssl_tls.c`.

References `_ssl_context::s_get`, and `_ssl_context::s_set`.

Referenced by `main()`.

**13.112.5.30 void ssl\_set\_session (ssl\_context \* ssl, int resume, int timeout, ssl\_session \* session)**

Set the session resuming flag, timeout and data.

**Parameters:**

*ssl* SSL context  
*resume* if 0 (default), the session will not be resumed  
*timeout* session timeout in seconds, or 0 (no timeout)  
*session* session context

Definition at line 1628 of file `ssl_tls.c`.

References `_ssl_context::resume`, `_ssl_context::session`, and `_ssl_context::timeout`.

Referenced by `main()`, and `ssl_test()`.

**13.112.5.31 int ssl\_write (ssl\_context \* ssl, unsigned char \* buf, int len)**

Write exactly 'len' application data bytes.

**Parameters:**

*ssl* SSL context  
*buf* buffer holding the data  
*len* how many bytes must be written

**Returns:**

This function returns the number of bytes written, or a negative error code.

**Note:**

When this function returns `XYSSL_ERR_NET_TRY_AGAIN`, it must be called later with the *\*same\** arguments, until it returns a positive value.

Definition at line 1857 of file `ssl_tls.c`.

References `_ssl_context::out_left`, `_ssl_context::out_msg`, `_ssl_context::out_msglen`, `_ssl_context::out_msgtype`, `SSL_DEBUG_MSG`, `SSL_DEBUG_RET`, `ssl_flush_output()`, `ssl_handshake()`, `SSL_HANDSHAKE_OVER`, `SSL_MAX_CONTENT_LEN`, `SSL_MSG_APPLICATION_DATA`, `ssl_write_record()`, and `_ssl_context::state`.

Referenced by `main()`, and `ssl_test()`.

### 13.112.5.32 `int ssl_write_certificate(ssl_context *ssl)`

Definition at line 1044 of file `ssl_tls.c`.

References `_ssl_context::client_auth`, `_ssl_context::endpoint`, `_x509_buf::len`, `_ssl_context::minor_ver`, `_x509_cert::next`, `_ssl_context::out_msg`, `_ssl_context::out_msglen`, `_ssl_context::out_msgtype`, `_ssl_context::own_cert`, `_x509_buf::p`, `_x509_cert::raw`, `SSL_ALERT_NO_CERTIFICATE`, `SSL_ALERT_WARNING`, `SSL_DEBUG_CRT`, `SSL_DEBUG_MSG`, `SSL_DEBUG_RET`, `SSL_HS_CERTIFICATE`, `SSL_IS_CLIENT`, `SSL_MAX_CONTENT_LEN`, `SSL_MINOR_VERSION_0`, `SSL_MSG_ALERT`, `SSL_MSG_HANDSHAKE`, `ssl_write_record()`, `_ssl_context::state`, `XYSSL_ERR_SSL_CERTIFICATE_REQUIRED`, and `XYSSL_ERR_SSL_CERTIFICATE_TOO_LARGE`.

Referenced by `ssl_handshake_client()`, and `ssl_handshake_server()`.

### 13.112.5.33 `int ssl_write_change_cipher_spec(ssl_context *ssl)`

Definition at line 1287 of file `ssl_tls.c`.

References `_ssl_context::do_crypt`, `_ssl_context::out_msg`, `_ssl_context::out_msglen`, `_ssl_context::out_msgtype`, `SSL_DEBUG_MSG`, `SSL_DEBUG_RET`, `SSL_MSG_CHANGE_CIPHER_SPEC`, `ssl_write_record()`, and `_ssl_context::state`.

Referenced by `ssl_handshake_client()`, and `ssl_handshake_server()`.

### 13.112.5.34 `int ssl_write_finished(ssl_context *ssl)`

Definition at line 1433 of file `ssl_tls.c`.

References `_ssl_context::do_crypt`, `_ssl_context::endpoint`, `_ssl_context::fin_md5`, `_ssl_context::fin_sha1`, `md5()`, `_ssl_context::minor_ver`, `_ssl_context::out_msg`, `_ssl_context::out_msglen`, `_ssl_context::out_msgtype`, `_ssl_context::resume`, `sha1()`, `ssl_calc_finished()`, `SSL_CLIENT_CHANGE_CIPHER_SPEC`, `SSL_DEBUG_MSG`, `SSL_DEBUG_RET`, `SSL_HANDSHAKE_OVER`, `SSL_HS_FINISHED`, `SSL_IS_CLIENT`, `SSL_MINOR_VERSION_0`, `SSL_MSG_HANDSHAKE`, `ssl_write_record()`, and `_ssl_context::state`.

Referenced by `ssl_handshake_client()`, and `ssl_handshake_server()`.

### 13.112.5.35 `int ssl_write_record(ssl_context *ssl)`

Definition at line 786 of file `ssl_tls.c`.

References `_ssl_context::do_crypt`, `_ssl_context::fin_md5`, `_ssl_context::fin_sha1`, `_ssl_context::major_ver`, `md5_update()`, `_ssl_context::minor_ver`, `_ssl_context::out_hdr`, `_ssl_context::out_left`, `_ssl_context::out_msg`, `_ssl_context::out_msglen`, `_ssl_context::out_msgtype`, `sha1_update()`, `SSL_DEBUG_BUF`, `SSL_DEBUG_MSG`, `SSL_DEBUG_RET`, `ssl_encrypt_buf()`, `ssl_flush_output()`, and `SSL_MSG_HANDSHAKE`.

Referenced by `ssl_close_notify()`, `ssl_write()`, `ssl_write_certificate()`, `ssl_write_certificate_request()`, `ssl_write_certificate_verify()`, `ssl_write_change_cipher_spec()`, `ssl_write_client_hello()`, `ssl_write_client_key_exchange()`, `ssl_write_finished()`, `ssl_write_server_hello()`, `ssl_write_server_hello_done()`, and `ssl_write_server_key_exchange()`.

### 13.112.6 Variable Documentation

#### 13.112.6.1 `int ssl_default_ciphers[ ]`

Definition at line 1739 of file `ssl_tls.c`.

Referenced by `main()`, and `ssl_test()`.



## Data Structures

- struct [hr\\_time](#)  
*timer structure*

## Functions

- unsigned long [hardclock](#) (void)  
*Return the CPU cycle counter value.*
- unsigned long [get\\_timer](#) (struct [hr\\_time](#) \*val, int reset)  
*Return the elapsed time in milliseconds.*
- void [set\\_alarm](#) (int seconds)  
*Setup an alarm clock.*
- void [m\\_sleep](#) (int milliseconds)  
*Sleep for a certain amount of time.*

## Variables

- int [alarmed](#)

### 13.113.1 Detailed Description

Definition in file [timing.h](#).

### 13.113.2 Function Documentation

#### 13.113.2.1 unsigned long [get\\_timer](#) (struct [hr\\_time](#) \* val, int reset)

Return the elapsed time in milliseconds.

##### Parameters:

*val* points to a timer structure  
*reset* if set to 1, the timer is restarted

Definition at line 205 of file [timing.c](#).

References [\\_hr\\_time::start](#).

Referenced by [ssl\\_test\(\)](#).

**13.113.2.2 unsigned long hardclock (void)**

Return the CPU cycle counter value.

Definition at line 136 of file timing.c.

References tv\_init.

Referenced by main().

**13.113.2.3 void m\_sleep (int *milliseconds*)**

Sleep for a certain amount of time.

Definition at line 238 of file timing.c.

**13.113.2.4 void set\_alarm (int *seconds*)**

Setup an alarm clock.

**Parameters:**

*seconds* delay before the "alarmed" flag is set

Definition at line 231 of file timing.c.

References sighandler().

Referenced by main().

**13.113.3 Variable Documentation****13.113.3.1 int alarmed**

Definition at line 159 of file timing.c.

Referenced by main().

## 13.114 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/x509.h File Reference

```
#include "xyssl/rsa.h"
```

### Data Structures

- struct [\\_x509\\_buf](#)
- struct [\\_x509\\_name](#)
- struct [\\_x509\\_time](#)
- struct [\\_x509\\_cert](#)
- struct [\\_x509\\_node](#)
- struct [\\_x509\\_raw](#)

### Defines

- #define [XYSSL\\_ERR\\_ASN1\\_OUT\\_OF\\_DATA](#) -0x0014
- #define [XYSSL\\_ERR\\_ASN1\\_UNEXPECTED\\_TAG](#) -0x0016
- #define [XYSSL\\_ERR\\_ASN1\\_INVALID\\_LENGTH](#) -0x0018
- #define [XYSSL\\_ERR\\_ASN1\\_LENGTH\\_MISMATCH](#) -0x001A
- #define [XYSSL\\_ERR\\_ASN1\\_INVALID\\_DATA](#) -0x001C
- #define [XYSSL\\_ERR\\_X509\\_FEATURE\\_UNAVAILABLE](#) -0x0020
- #define [XYSSL\\_ERR\\_X509\\_CERT\\_INVALID\\_PEM](#) -0x0040
- #define [XYSSL\\_ERR\\_X509\\_CERT\\_INVALID\\_FORMAT](#) -0x0060
- #define [XYSSL\\_ERR\\_X509\\_CERT\\_INVALID\\_VERSION](#) -0x0080
- #define [XYSSL\\_ERR\\_X509\\_CERT\\_INVALID\\_SERIAL](#) -0x00A0
- #define [XYSSL\\_ERR\\_X509\\_CERT\\_INVALID\\_ALG](#) -0x00C0
- #define [XYSSL\\_ERR\\_X509\\_CERT\\_INVALID\\_NAME](#) -0x00E0
- #define [XYSSL\\_ERR\\_X509\\_CERT\\_INVALID\\_DATE](#) -0x0100
- #define [XYSSL\\_ERR\\_X509\\_CERT\\_INVALID\\_PUBKEY](#) -0x0120
- #define [XYSSL\\_ERR\\_X509\\_CERT\\_INVALID\\_SIGNATURE](#) -0x0140
- #define [XYSSL\\_ERR\\_X509\\_CERT\\_INVALID\\_EXTENSIONS](#) -0x0160
- #define [XYSSL\\_ERR\\_X509\\_CERT\\_UNKNOWN\\_VERSION](#) -0x0180
- #define [XYSSL\\_ERR\\_X509\\_CERT\\_UNKNOWN\\_SIG\\_ALG](#) -0x01A0
- #define [XYSSL\\_ERR\\_X509\\_CERT\\_UNKNOWN\\_PK\\_ALG](#) -0x01C0
- #define [XYSSL\\_ERR\\_X509\\_CERT\\_SIG\\_MISMATCH](#) -0x01E0
- #define [XYSSL\\_ERR\\_X509\\_CERT\\_VERIFY\\_FAILED](#) -0x0200
- #define [XYSSL\\_ERR\\_X509\\_KEY\\_INVALID\\_PEM](#) -0x0220
- #define [XYSSL\\_ERR\\_X509\\_KEY\\_INVALID\\_VERSION](#) -0x0240
- #define [XYSSL\\_ERR\\_X509\\_KEY\\_INVALID\\_FORMAT](#) -0x0260
- #define [XYSSL\\_ERR\\_X509\\_KEY\\_INVALID\\_ENC\\_IV](#) -0x0280
- #define [XYSSL\\_ERR\\_X509\\_KEY\\_UNKNOWN\\_ENC\\_ALG](#) -0x02A0
- #define [XYSSL\\_ERR\\_X509\\_KEY\\_PASSWORD\\_REQUIRED](#) -0x02C0
- #define [XYSSL\\_ERR\\_X509\\_KEY\\_PASSWORD\\_MISMATCH](#) -0x02E0
- #define [XYSSL\\_ERR\\_X509\\_POINT\\_ERROR](#) -0x0300
- #define [XYSSL\\_ERR\\_X509\\_VALUE\\_TO\\_LENGTH](#) -0x0320
- #define [BADCERT\\_EXPIRED](#) 1

- #define [BADCERT\\_REVOKED](#) 2
- #define [BADCERT\\_CN\\_MISMATCH](#) 4
- #define [BADCERT\\_NOT\\_TRUSTED](#) 8
- #define [ASN1\\_BOOLEAN](#) 0x01
- #define [ASN1\\_INTEGER](#) 0x02
- #define [ASN1\\_BIT\\_STRING](#) 0x03
- #define [ASN1\\_OCTET\\_STRING](#) 0x04
- #define [ASN1\\_NULL](#) 0x05
- #define [ASN1\\_OID](#) 0x06
- #define [ASN1\\_UTF8\\_STRING](#) 0x0C
- #define [ASN1\\_SEQUENCE](#) 0x10
- #define [ASN1\\_SET](#) 0x11
- #define [ASN1\\_PRINTABLE\\_STRING](#) 0x13
- #define [ASN1\\_T61\\_STRING](#) 0x14
- #define [ASN1\\_IA5\\_STRING](#) 0x16
- #define [ASN1\\_UTC\\_TIME](#) 0x17
- #define [ASN1\\_UNIVERSAL\\_STRING](#) 0x1C
- #define [ASN1\\_BMP\\_STRING](#) 0x1E
- #define [ASN1\\_PRIMITIVE](#) 0x00
- #define [ASN1\\_CONSTRUCTED](#) 0x20
- #define [ASN1\\_CONTEXT\\_SPECIFIC](#) 0x80
- #define [X520\\_COMMON\\_NAME](#) 3
- #define [X520\\_COUNTRY](#) 6
- #define [X520\\_LOCALITY](#) 7
- #define [X520\\_STATE](#) 8
- #define [X520\\_ORGANIZATION](#) 10
- #define [X520\\_ORG\\_UNIT](#) 11
- #define [PKCS9\\_EMAIL](#) 1
- #define [X509\\_OUTPUT\\_DER](#) 0x01
- #define [X509\\_OUTPUT\\_PEM](#) 0x02
- #define [PEM\\_LINE\\_LENGTH](#) 72
- #define [X509\\_ISSUER](#) 0x01
- #define [X509\\_SUBJECT](#) 0x02
- #define [OID\\_X520](#) "\x55\x04"
- #define [OID\\_CN](#) "\x55\x04\x03"
- #define [OID\\_PKCS1](#) "\x2A\x86\x48\x86\xF7\x0D\x01\x01"
- #define [OID\\_PKCS1\\_RSA](#) "\x2A\x86\x48\x86\xF7\x0D\x01\x01\x01"
- #define [OID\\_PKCS1\\_RSA\\_SHA](#) "\x2A\x86\x48\x86\xF7\x0D\x01\x01\x05"
- #define [OID\\_PKCS9](#) "\x2A\x86\x48\x86\xF7\x0D\x01\x09"
- #define [OID\\_PKCS9\\_EMAIL](#) "\x2A\x86\x48\x86\xF7\x0D\x01\x09\x01"

## Typedefs

- typedef struct [\\_x509\\_buf](#) [x509\\_buf](#)
- typedef struct [\\_x509\\_name](#) [x509\\_name](#)
- typedef struct [\\_x509\\_time](#) [x509\\_time](#)
- typedef struct [\\_x509\\_cert](#) [x509\\_cert](#)
- typedef struct [\\_x509\\_node](#) [x509\\_node](#)
- typedef struct [\\_x509\\_raw](#) [x509\\_raw](#)

**Functions**

- `int x509parse_cert` (`x509_cert` \*crt, unsigned char \*buf, int buflen)  
*Parse one or more certificates and add them to the chained list.*
- `int x509parse_crtfile` (`x509_cert` \*crt, char \*path)  
*Load one or more certificates and add them to the chained list.*
- `int x509parse_key` (`rsa_context` \*rsa, unsigned char \*buf, int buflen, unsigned char \*pwd, int pwrlen)  
*Parse a private RSA key.*
- `int x509parse_keyfile` (`rsa_context` \*rsa, char \*path, char \*password)  
*Load and parse a private RSA key.*
- `int x509parse_dn_gets` (char \*buf, char \*end, `x509_name` \*dn)  
*Store the certificate DN in printable form into buf; no more than (end - buf) characters will be written.*
- `char * x509parse_cert_info` (char \*prefix, `x509_cert` \*crt)  
*Returns an informational string about the certificate.*
- `int x509parse_expired` (`x509_cert` \*crt)  
*Return 0 if the certificate is still valid, or BADCERT\_EXPIRED.*
- `int x509parse_verify` (`x509_cert` \*crt, `x509_cert` \*trust\_ca, char \*cn, int \*flags)  
*Verify the certificate signature.*
- `void x509_free` (`x509_cert` \*crt)  
*Unallocate all certificate data.*
- `int x509_self_test` (int verbose)  
*Checkup routine.*

**13.114.1 Detailed Description**

Definition in file [x509.h](#).

**13.114.2 Define Documentation****13.114.2.1 #define ASN1\_BIT\_STRING 0x03**

Definition at line 51 of file [x509.h](#).

Referenced by [x509\\_get\\_pubkey\(\)](#), and [x509\\_get\\_sig\(\)](#).

**13.114.2.2 #define ASN1\_BMP\_STRING 0x1E**

Definition at line 63 of file [x509.h](#).

Referenced by [x509\\_get\\_name\(\)](#).

**13.114.2.3 #define ASN1\_BOOLEAN 0x01**

Definition at line 49 of file x509.h.

Referenced by `asn1_get_bool()`.

**13.114.2.4 #define ASN1\_CONSTRUCTED 0x20**

Definition at line 65 of file x509.h.

Referenced by `x509_get_alg()`, `x509_get_dates()`, `x509_get_ext()`, `x509_get_name()`, `x509_get_pubkey()`, `x509_get_uid()`, `x509_get_version()`, `x509parse_crt()`, and `x509parse_key()`.

**13.114.2.5 #define ASN1\_CONTEXT\_SPECIFIC 0x80**

Definition at line 66 of file x509.h.

Referenced by `x509_get_ext()`, `x509_get_serial()`, `x509_get_uid()`, and `x509_get_version()`.

**13.114.2.6 #define ASN1\_IA5\_STRING 0x16**

Definition at line 60 of file x509.h.

Referenced by `x509_get_name()`.

**13.114.2.7 #define ASN1\_INTEGER 0x02**

Definition at line 50 of file x509.h.

Referenced by `asn1_get_int()`, `asn1_get_mpi()`, and `x509_get_serial()`.

**13.114.2.8 #define ASN1\_NULL 0x05**

Definition at line 53 of file x509.h.

Referenced by `x509_get_alg()`.

**13.114.2.9 #define ASN1\_OCTET\_STRING 0x04**

Definition at line 52 of file x509.h.

Referenced by `x509_get_ext()`.

**13.114.2.10 #define ASN1\_OID 0x06**

Definition at line 54 of file x509.h.

Referenced by `x509_get_alg()`, and `x509_get_name()`.

**13.114.2.11 #define ASN1\_PRIMITIVE 0x00**

Definition at line 64 of file x509.h.

### 13.114

/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/x509.h File

#### Reference

763

Referenced by x509\_get\_serial().

---

#### 13.114.2.12 #define ASN1\_PRINTABLE\_STRING 0x13

Definition at line 58 of file x509.h.

Referenced by x509\_get\_name().

#### 13.114.2.13 #define ASN1\_SEQUENCE 0x10

Definition at line 56 of file x509.h.

Referenced by x509\_get\_alg(), x509\_get\_dates(), x509\_get\_ext(), x509\_get\_name(), x509\_get\_pubkey(), x509parse\_crt(), and x509parse\_key().

#### 13.114.2.14 #define ASN1\_SET 0x11

Definition at line 57 of file x509.h.

Referenced by x509\_get\_name().

#### 13.114.2.15 #define ASN1\_T61\_STRING 0x14

Definition at line 59 of file x509.h.

Referenced by x509\_get\_name().

#### 13.114.2.16 #define ASN1\_UNIVERSAL\_STRING 0x1C

Definition at line 62 of file x509.h.

Referenced by x509\_get\_name().

#### 13.114.2.17 #define ASN1.UTC\_TIME 0x17

Definition at line 61 of file x509.h.

Referenced by x509\_get\_dates().

#### 13.114.2.18 #define ASN1\_UTF8\_STRING 0x0C

Definition at line 55 of file x509.h.

Referenced by x509\_get\_name().

#### 13.114.2.19 #define BADCERT\_CN\_MISMATCH 4

Definition at line 43 of file x509.h.

Referenced by main(), and x509parse\_verify().

**13.114.2.20 #define BADCERT\_EXPIRED 1**

Definition at line 41 of file x509.h.

Referenced by main(), and x509parse\_expired().

**13.114.2.21 #define BADCERT\_NOT\_TRUSTED 8**

Definition at line 44 of file x509.h.

Referenced by main(), and x509parse\_verify().

**13.114.2.22 #define BADCERT\_REVOKED 2**

Definition at line 42 of file x509.h.

Referenced by main().

**13.114.2.23 #define OID\_CN "\x55\x04\x03"**

Definition at line 86 of file x509.h.

Referenced by x509parse\_verify().

**13.114.2.24 #define OID\_PKCS1 "\x2A\x86\x48\x86\xF7\x0D\x01\x01"**

Definition at line 87 of file x509.h.

Referenced by x509parse\_crt().

**13.114.2.25 #define OID\_PKCS1\_RSA "\x2A\x86\x48\x86\xF7\x0D\x01\x01\x01"**

Definition at line 88 of file x509.h.

Referenced by x509\_get\_pubkey().

**13.114.2.26 #define OID\_PKCS1\_RSA\_SHA "\x2A\x86\x48\x86\xF7\x0D\x01\x01\x05"**

Definition at line 89 of file x509.h.

**13.114.2.27 #define OID\_PKCS9 "\x2A\x86\x48\x86\xF7\x0D\x01\x09"**

Definition at line 90 of file x509.h.

Referenced by x509parse\_dn\_gets().

**13.114.2.28 #define OID\_PKCS9\_EMAIL "\x2A\x86\x48\x86\xF7\x0D\x01\x09\x01"**

Definition at line 91 of file x509.h.



### 13.114

/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/x509.h File Reference 765

**13.114.2.29 #define OID\_X520 "\x55\x04"**

Definition at line 85 of file x509.h.

Referenced by x509parse\_dn\_gets().

**13.114.2.30 #define PEM\_LINE\_LENGTH 72**

Definition at line 81 of file x509.h.

**13.114.2.31 #define PKCS9\_EMAIL 1**

Definition at line 77 of file x509.h.

Referenced by x509parse\_dn\_gets().

**13.114.2.32 #define X509\_ISSUER 0x01**

Definition at line 82 of file x509.h.

**13.114.2.33 #define X509\_OUTPUT\_DER 0x01**

Definition at line 79 of file x509.h.

**13.114.2.34 #define X509\_OUTPUT\_PEM 0x02**

Definition at line 80 of file x509.h.

**13.114.2.35 #define X509\_SUBJECT 0x02**

Definition at line 83 of file x509.h.

**13.114.2.36 #define X520\_COMMON\_NAME 3**

Definition at line 71 of file x509.h.

Referenced by x509parse\_dn\_gets().

**13.114.2.37 #define X520\_COUNTRY 6**

Definition at line 72 of file x509.h.

Referenced by x509parse\_dn\_gets().

**13.114.2.38 #define X520\_LOCALITY 7**

Definition at line 73 of file x509.h.

Referenced by x509parse\_dn\_gets().

**13.114.2.39 #define X520\_ORG\_UNIT 11**

Definition at line 76 of file x509.h.

Referenced by x509parse\_dn\_gets().

**13.114.2.40 #define X520\_ORGANIZATION 10**

Definition at line 75 of file x509.h.

Referenced by x509parse\_dn\_gets().

**13.114.2.41 #define X520\_STATE 8**

Definition at line 74 of file x509.h.

Referenced by x509parse\_dn\_gets().

**13.114.2.42 #define XYSSL\_ERR\_ASN1\_INVALID\_DATA -0x001C**

Definition at line 13 of file x509.h.

**13.114.2.43 #define XYSSL\_ERR\_ASN1\_INVALID\_LENGTH -0x0018**

Definition at line 11 of file x509.h.

Referenced by asn1\_get\_bool(), asn1\_get\_int(), and asn1\_get\_len().

**13.114.2.44 #define XYSSL\_ERR\_ASN1\_LENGTH\_MISMATCH -0x001A**

Definition at line 12 of file x509.h.

Referenced by x509\_get\_alg(), x509\_get\_dates(), x509\_get\_ext(), x509\_get\_name(), x509\_get\_pubkey(), x509\_get\_version(), x509parse\_crt(), and x509parse\_key().

**13.114.2.45 #define XYSSL\_ERR\_ASN1\_OUT\_OF\_DATA -0x0014**

Definition at line 9 of file x509.h.

Referenced by asn1\_get\_len(), asn1\_get\_tag(), x509\_get\_name(), x509\_get\_pubkey(), and x509\_get\_serial().

**13.114.2.46 #define XYSSL\_ERR\_ASN1\_UNEXPECTED\_TAG -0x0016**

Definition at line 10 of file x509.h.

Referenced by asn1\_get\_tag(), x509\_get\_ext(), x509\_get\_name(), x509\_get\_serial(), x509\_get\_uid(), and x509\_get\_version().

### 13.114

/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/x509.h File Reference 767

**13.114.2.47 #define XYSSL\_ERR\_X509\_CERT\_INVALID\_ALG -0x00C0**

Definition at line 20 of file x509.h.

Referenced by x509\_get\_alg().

**13.114.2.48 #define XYSSL\_ERR\_X509\_CERT\_INVALID\_DATE -0x0100**

Definition at line 22 of file x509.h.

Referenced by x509\_get\_dates().

**13.114.2.49 #define XYSSL\_ERR\_X509\_CERT\_INVALID\_EXTENSIONS -0x0160**

Definition at line 25 of file x509.h.

Referenced by x509\_get\_ext().

**13.114.2.50 #define XYSSL\_ERR\_X509\_CERT\_INVALID\_FORMAT -0x0060**

Definition at line 17 of file x509.h.

Referenced by x509parse\_crt().

**13.114.2.51 #define XYSSL\_ERR\_X509\_CERT\_INVALID\_NAME -0x00E0**

Definition at line 21 of file x509.h.

Referenced by x509\_get\_name().

**13.114.2.52 #define XYSSL\_ERR\_X509\_CERT\_INVALID\_PEM -0x0040**

Definition at line 16 of file x509.h.

Referenced by x509parse\_crt().

**13.114.2.53 #define XYSSL\_ERR\_X509\_CERT\_INVALID\_PUBKEY -0x0120**

Definition at line 23 of file x509.h.

Referenced by x509\_get\_pubkey().

**13.114.2.54 #define XYSSL\_ERR\_X509\_CERT\_INVALID\_SERIAL -0x00A0**

Definition at line 19 of file x509.h.

Referenced by x509\_get\_serial().

**13.114.2.55 #define XYSSL\_ERR\_X509\_CERT\_INVALID\_SIGNATURE -0x0140**

Definition at line 24 of file x509.h.

Referenced by x509\_get\_sig().

**13.114.2.56 #define XYSSL\_ERR\_X509\_CERT\_INVALID\_VERSION -0x0080**

Definition at line 18 of file x509.h.

Referenced by x509\_get\_version().

**13.114.2.57 #define XYSSL\_ERR\_X509\_CERT\_SIG\_MISMATCH -0x01E0**

Definition at line 29 of file x509.h.

Referenced by x509parse\_crt().

**13.114.2.58 #define XYSSL\_ERR\_X509\_CERT\_UNKNOWN\_PK\_ALG -0x01C0**

Definition at line 28 of file x509.h.

Referenced by x509\_get\_pubkey().

**13.114.2.59 #define XYSSL\_ERR\_X509\_CERT\_UNKNOWN\_SIG\_ALG -0x01A0**

Definition at line 27 of file x509.h.

Referenced by x509parse\_crt().

**13.114.2.60 #define XYSSL\_ERR\_X509\_CERT\_UNKNOWN\_VERSION -0x0180**

Definition at line 26 of file x509.h.

Referenced by x509parse\_crt().

**13.114.2.61 #define XYSSL\_ERR\_X509\_CERT\_VERIFY\_FAILED -0x0200**

Definition at line 30 of file x509.h.

Referenced by x509parse\_verify().

**13.114.2.62 #define XYSSL\_ERR\_X509\_FEATURE\_UNAVAILABLE -0x0020**

Definition at line 15 of file x509.h.

Referenced by x509parse\_key().

**13.114.2.63 #define XYSSL\_ERR\_X509\_KEY\_INVALID\_ENC\_IV -0x0280**

Definition at line 34 of file x509.h.

Referenced by x509\_get\_iv(), and x509parse\_key().

**13.114.2.64 #define XYSSL\_ERR\_X509\_KEY\_INVALID\_FORMAT -0x0260**

Definition at line 33 of file x509.h.

Referenced by x509parse\_key().

### 13.114

/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/x509.h File Reference 769

**13.114.2.65** `#define XYSSL_ERR_X509_KEY_INVALID_PEM -0x0220`

Definition at line 31 of file x509.h.

Referenced by x509parse\_key().

**13.114.2.66** `#define XYSSL_ERR_X509_KEY_INVALID_VERSION -0x0240`

Definition at line 32 of file x509.h.

Referenced by x509parse\_key().

**13.114.2.67** `#define XYSSL_ERR_X509_KEY_PASSWORD_MISMATCH -0x02E0`

Definition at line 37 of file x509.h.

Referenced by x509parse\_key().

**13.114.2.68** `#define XYSSL_ERR_X509_KEY_PASSWORD_REQUIRED -0x02C0`

Definition at line 36 of file x509.h.

Referenced by x509parse\_key().

**13.114.2.69** `#define XYSSL_ERR_X509_KEY_UNKNOWN_ENC_ALG -0x02A0`

Definition at line 35 of file x509.h.

Referenced by x509parse\_key().

**13.114.2.70** `#define XYSSL_ERR_X509_POINT_ERROR -0x0300`

Definition at line 38 of file x509.h.

**13.114.2.71** `#define XYSSL_ERR_X509_VALUE_TO_LENGTH -0x0320`

Definition at line 39 of file x509.h.

### 13.114.3 Typedef Documentation

**13.114.3.1** `typedef struct _x509_buf x509_buf`

**13.114.3.2** `typedef struct _x509_cert x509_cert`

**13.114.3.3** `typedef struct _x509_name x509_name`

**13.114.3.4** `typedef struct _x509_node x509_node`

**13.114.3.5** `typedef struct _x509_raw x509_raw`

**13.114.3.6** `typedef struct _x509_time x509_time`

### 13.114.4 Function Documentation

**13.114.4.1** `void x509_free (x509_cert * crt)`

Unallocate all certificate data.

Definition at line 1613 of file x509parse.c.

References `_x509_cert::issuer`, `_x509_buf::len`, `_x509_cert::next`, `_x509_name::next`, `_x509_buf::p`, `_x509_cert::raw`, `_x509_cert::rsa`, `rsa_free()`, and `_x509_cert::subject`.

Referenced by `main()`, `ssl_free()`, `ssl_test()`, `x509_self_test()`, and `x509parse_cert()`.

**13.114.4.2** `int x509_self_test (int verbose)`

Checkup routine.

#### Returns:

0 if successful, or 1 if the test failed

Definition at line 1675 of file x509parse.c.

References `rsa_free()`, `test_ca_cert`, `test_ca_key`, `test_ca_pwd`, `test_cli_cert`, `x509_free()`, `x509parse_cert()`, `x509parse_key()`, and `x509parse_verify()`.

Referenced by `main()`.

**13.114.4.3** `char* x509parse_cert_info (char * prefix, x509_cert * crt)`

Returns an informational string about the certificate.

Definition at line 1399 of file x509parse.c.

References `buf`, `_x509_time::day`, `_x509_time::hour`, `_x509_cert::issuer`, `_x509_buf::len`, `_x509_time::min`, `_x509_time::mon`, `mpi::n`, `rsa_context::N`, `_x509_buf::p`, `_x509_cert::rsa`, `RSA_MD2`, `RSA_MD4`, `RSA_MD5`, `RSA_SHA1`, `_x509_time::sec`, `_x509_cert::serial`, `_x509_cert::sig_oid1`, `_x509_cert::subject`, `_x509_cert::valid_from`, `_x509_cert::valid_to`, `_x509_cert::version`, `x509parse_dn_gets()`, and `_x509_time::year`.

Referenced by `debug_print_cert()`, and `main()`.

### 13.114

/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/x509.h File Reference 771

#### 13.114.4.4 `int x509parse_crt(x509_cert *crt, unsigned char *buf, int buflen)`

Parse one or more certificates and add them to the chained list.

##### Parameters:

*chain* points to the start of the chain  
*buf* buffer holding the certificate data  
*buflen* size of the buffer

##### Returns:

0 if successful, or a specific X509 error code

Definition at line 647 of file x509parse.c.

References ASN1\_CONSTRUCTED, asn1\_get\_tag(), ASN1\_SEQUENCE, base64\_decode(), \_x509\_cert::ca\_istrue, rsa\_context::E, \_x509\_cert::issuer, \_x509\_cert::issuer\_id, \_x509\_cert::issuer\_raw, rsa\_context::len, \_x509\_buf::len, \_x509\_cert::max\_pathlen, mpi\_size(), rsa\_context::N, \_x509\_cert::next, OID\_PKCS1, \_x509\_buf::p, \_x509\_cert::pk\_oid, \_x509\_cert::raw, \_x509\_cert::rsa, rsa\_check\_pubkey(), \_x509\_cert::serial, \_x509\_cert::sig, \_x509\_cert::sig\_oid1, \_x509\_cert::sig\_oid2, \_x509\_cert::subject, \_x509\_cert::subject\_id, \_x509\_cert::subject\_raw, \_x509\_cert::tbs, \_x509\_cert::v3\_ext, \_x509\_cert::valid\_from, \_x509\_cert::valid\_to, \_x509\_cert::version, x509\_free(), x509\_get\_alg(), x509\_get\_dates(), x509\_get\_ext(), x509\_get\_name(), x509\_get\_pubkey(), x509\_get\_serial(), x509\_get\_sig(), x509\_get\_uid(), x509\_get\_version(), x509parse\_crt(), XYSSL\_ERR\_ASN1\_LENGTH\_MISMATCH, XYSSL\_ERR\_BASE64\_INVALID\_CHARACTER, XYSSL\_ERR\_X509\_CERT\_INVALID\_FORMAT, XYSSL\_ERR\_X509\_CERT\_INVALID\_PEM, XYSSL\_ERR\_X509\_CERT\_SIG\_MISMATCH, XYSSL\_ERR\_X509\_CERT\_UNKNOWN\_SIG\_ALG, and XYSSL\_ERR\_X509\_CERT\_UNKNOWN\_VERSION.

Referenced by main(), ssl\_parse\_certificate(), ssl\_test(), x509\_self\_test(), x509parse\_crt(), and x509parse\_crtfile().

#### 13.114.4.5 `int x509parse_crtfile(x509_cert *crt, char *path)`

Load one or more certificates and add them to the chained list.

##### Parameters:

*chain* points to the start of the chain  
*path* filename to read the certificates from

##### Returns:

0 if successful, or a specific X509 error code

Definition at line 979 of file x509parse.c.

References buf, f, and x509parse\_crt().

#### 13.114.4.6 `int x509parse_dn_gets(char *buf, char *end, x509_name *dn)`

Store the certificate DN in printable form into buf; no more than (end - buf) characters will be written.

Definition at line 1316 of file x509parse.c.

References `_x509_buf::len`, `_x509_name::next`, `_x509_name::oid`, `OID_PKCS9`, `OID_X520`, `_x509_buf::p`, `PKCS9_EMAIL`, `_x509_name::val`, `X520_COMMON_NAME`, `X520_COUNTRY`, `X520_LOCALITY`, `X520_ORG_UNIT`, `X520_ORGANIZATION`, and `X520_STATE`.

Referenced by `x509parse_cert_info()`.

#### 13.114.4.7 `int x509parse_expired(x509_cert * crt)`

Return 0 if the certificate is still valid, or `BADCERT_EXPIRED`.

Definition at line 1458 of file `x509parse.c`.

References `BADCERT_EXPIRED`, `_x509_time::day`, `_x509_time::mon`, `_x509_cert::valid_to`, and `_x509_time::year`.

Referenced by `x509parse_verify()`.

#### 13.114.4.8 `int x509parse_key(rsa_context * rsa, unsigned char * buf, int buflen, unsigned char * pwd, int pwrlen)`

Parse a private RSA key.

##### Parameters:

*rsa* RSA context to be initialized  
*buf* input buffer  
*buflen* size of the buffer  
*pwd* password for decryption (optional)  
*pwrlen* size of the password

##### Returns:

0 if successful, or a specific X509 error code

Definition at line 1082 of file `x509parse.c`.

References `ASN1_CONSTRUCTED`, `asn1_get_int()`, `asn1_get_mpi()`, `asn1_get_tag()`, `ASN1_SEQUENCE`, `base64_decode()`, `rsa_context::D`, `rsa_context::DP`, `rsa_context::DQ`, `rsa_context::E`, `rsa_context::len`, `mpi_size()`, `rsa_context::N`, `rsa_context::P`, `rsa_context::Q`, `rsa_context::QP`, `rsa_check_privkey()`, `rsa_free()`, `rsa_context::ver`, `x509_des3_decrypt()`, `x509_get_iv()`, `XYSSL_ERR_ASN1_LENGTH_MISMATCH`, `XYSSL_ERR_BASE64_INVALID_CHARACTER`, `XYSSL_ERR_X509_FEATURE_UNAVAILABLE`, `XYSSL_ERR_X509_KEY_INVALID_ENC_IV`, `XYSSL_ERR_X509_KEY_INVALID_FORMAT`, `XYSSL_ERR_X509_KEY_INVALID_PEM`, `XYSSL_ERR_X509_KEY_INVALID_VERSION`, `XYSSL_ERR_X509_KEY_PASSWORD_MISMATCH`, `XYSSL_ERR_X509_KEY_PASSWORD_REQUIRED`, and `XYSSL_ERR_X509_KEY_UNKNOWN_ENC_ALG`.

Referenced by `main()`, `ssl_test()`, `x509_self_test()`, and `x509parse_keyfile()`.

#### 13.114.4.9 `int x509parse_keyfile(rsa_context * rsa, char * path, char * password)`

Load and parse a private RSA key.

##### Parameters:

*rsa* RSA context to be initialized



### 13.114

/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/include/xyssl/x509.h File Reference 773

*path* filename to read the private key from

*pwd* password to decrypt the file (can be NULL)

#### Returns:

0 if successful, or a specific X509 error code

Definition at line 1269 of file x509parse.c.

References buf, f, and x509parse\_key().

#### 13.114.4.10 int x509parse\_verify (x509\_cert \* crt, x509\_cert \* trust\_ca, char \* cn, int \* flags)

Verify the certificate signature.

#### Parameters:

*crt* a certificate to be verified

*trust\_ca* the trusted CA chain

*cn* expected Common Name (can be set to NULL if the CN must not be verified)

*flags* result of the verification

#### Returns:

0 if successful or XYSSL\_ERR\_X509\_SIG\_VERIFY\_FAILED, in which case \*flags will have one or more of the following values set: BADCERT\_EXPIRED -- BADCERT\_REVOKED -- BADCERT\_CN\_MISMATCH -- BADCERT\_NOT\_TRUSTED

#### Note:

TODO: add two arguments, depth and crl

Definition at line 1503 of file x509parse.c.

References BADCERT\_CN\_MISMATCH, BADCERT\_NOT\_TRUSTED, \_x509\_cert::ca\_istrue, cur, \_x509\_cert::issuer\_raw, \_x509\_buf::len, \_x509\_cert::max\_pathlen, \_x509\_cert::next, \_x509\_name::next, \_x509\_name::oid, OID\_CN, \_x509\_buf::p, \_x509\_cert::rsa, rsa\_pkcs1\_verify(), RSA\_PUBLIC, \_x509\_cert::sig, \_x509\_cert::sig\_oid1, \_x509\_cert::subject, \_x509\_cert::subject\_raw, \_x509\_cert::tbs, \_x509\_name::val, \_x509\_cert::version, x509\_hash(), x509parse\_expired(), and XYSSL\_ERR\_X509\_CERT\_VERIFY\_FAILED.

Referenced by ssl\_parse\_certificate(), and x509\_self\_test().

## 13.115 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/library/aes.c File Reference

```
#include "xyssl/config.h"
#include "xyssl/aes.h"
#include "xyssl/padlock.h"
#include <string.h>
#include <stdio.h>
```

### Defines

- #define [GET\\_ULONG\\_LE](#)(n, b, i)
- #define [PUT\\_ULONG\\_LE](#)(n, b, i)
- #define [ROTL8](#)(x) ( ( x << 8 ) & 0xFFFFFFFF ) | ( x >> 24 )
- #define [XTIME](#)(x) ( ( x << 1 ) ^ ( ( x & 0x80 ) ? 0x1B : 0x00 ) )
- #define [MUL](#)(x, y) ( ( x && y ) ? pow[(log[x]+log[y]) % 255] : 0 )
- #define [AES\\_FROUND](#)(X0, X1, X2, X3, Y0, Y1, Y2, Y3)
- #define [AES\\_RROUND](#)(X0, X1, X2, X3, Y0, Y1, Y2, Y3)

### Functions

- static void [aes\\_gen\\_tables](#) (void)
- void [aes\\_setkey\\_enc](#) ([aes\\_context](#) \*ctx, unsigned char \*key, [int](#) keysize)  
*AES key schedule (encryption).*
- void [aes\\_setkey\\_dec](#) ([aes\\_context](#) \*ctx, unsigned char \*key, [int](#) keysize)  
*AES key schedule (decryption).*
- void [aes\\_crypt\\_ecb](#) ([aes\\_context](#) \*ctx, [int](#) mode, unsigned char input[16], unsigned char output[16])  
*AES-ECB block encryption/decryption.*
- void [aes\\_crypt\\_cbc](#) ([aes\\_context](#) \*ctx, [int](#) mode, [int](#) length, unsigned char iv[16], unsigned char \*input, unsigned char \*output)  
*AES-CBC buffer encryption/decryption.*
- void [aes\\_crypt\\_cfb](#) ([aes\\_context](#) \*ctx, [int](#) mode, [int](#) length, [int](#) \*iv\_off, unsigned char iv[16], unsigned char \*input, unsigned char \*output)  
*AES-CFB buffer encryption/decryption.*
- [int](#) [aes\\_self\\_test](#) ([int](#) verbose)  
*Checkup routine.*

## Variables

- static unsigned char [FSb](#) [256]
- static unsigned long [FT0](#) [256]
- static unsigned long [FT1](#) [256]
- static unsigned long [FT2](#) [256]
- static unsigned long [FT3](#) [256]
- static unsigned char [RSb](#) [256]
- static unsigned long [RT0](#) [256]
- static unsigned long [RT1](#) [256]
- static unsigned long [RT2](#) [256]
- static unsigned long [RT3](#) [256]
- static unsigned long [RCON](#) [10]
- static `int` [aes\\_init\\_done](#) = 0
- static const unsigned char [aes\\_test\\_ecb\\_dec](#) [3][16]
- static const unsigned char [aes\\_test\\_ecb\\_enc](#) [3][16]
- static const unsigned char [aes\\_test\\_cbc\\_dec](#) [3][16]
- static const unsigned char [aes\\_test\\_cbc\\_enc](#) [3][16]
- static const unsigned char [aes\\_test\\_cfb\\_dec](#) [3][16]
- static const unsigned char [aes\\_test\\_cfb\\_enc](#) [3][16]

### 13.115.1 Define Documentation

#### 13.115.1.1 `#define` AES\_FROUND(*X0*, *X1*, *X2*, *X3*, *Y0*, *Y1*, *Y2*, *Y3*)

**Value:**

```
{
    X0 = *RK++ ^ FT0[ ( Y0      ) & 0xFF ] ^ \
              FT1[ ( Y1 >>  8 ) & 0xFF ] ^ \
              FT2[ ( Y2 >> 16 ) & 0xFF ] ^ \
              FT3[ ( Y3 >> 24 ) & 0xFF ]; \
    X1 = *RK++ ^ FT0[ ( Y1      ) & 0xFF ] ^ \
              FT1[ ( Y2 >>  8 ) & 0xFF ] ^ \
              FT2[ ( Y3 >> 16 ) & 0xFF ] ^ \
              FT3[ ( Y0 >> 24 ) & 0xFF ]; \
    X2 = *RK++ ^ FT0[ ( Y2      ) & 0xFF ] ^ \
              FT1[ ( Y3 >>  8 ) & 0xFF ] ^ \
              FT2[ ( Y0 >> 16 ) & 0xFF ] ^ \
              FT3[ ( Y1 >> 24 ) & 0xFF ]; \
    X3 = *RK++ ^ FT0[ ( Y3      ) & 0xFF ] ^ \
              FT1[ ( Y0 >>  8 ) & 0xFF ] ^ \
              FT2[ ( Y1 >> 16 ) & 0xFF ] ^ \
              FT3[ ( Y2 >> 24 ) & 0xFF ]; \
}
```

Definition at line 590 of file `aes.c`.

Referenced by `aes_crypt_ecb()`.

**13.115.1.2 #define AES\_RROUND(X0, X1, X2, X3, Y0, Y1, Y2, Y3)****Value:**

```

{
    X0 = *RK++ ^ RT0[ ( Y0          ) & 0xFF ] ^ \
                RT1[ ( Y3 >>  8 ) & 0xFF ] ^ \
                RT2[ ( Y2 >> 16 ) & 0xFF ] ^ \
                RT3[ ( Y1 >> 24 ) & 0xFF ]; \
    X1 = *RK++ ^ RT0[ ( Y1          ) & 0xFF ] ^ \
                RT1[ ( Y0 >>  8 ) & 0xFF ] ^ \
                RT2[ ( Y3 >> 16 ) & 0xFF ] ^ \
                RT3[ ( Y2 >> 24 ) & 0xFF ]; \
    X2 = *RK++ ^ RT0[ ( Y2          ) & 0xFF ] ^ \
                RT1[ ( Y1 >>  8 ) & 0xFF ] ^ \
                RT2[ ( Y0 >> 16 ) & 0xFF ] ^ \
                RT3[ ( Y3 >> 24 ) & 0xFF ]; \
    X3 = *RK++ ^ RT0[ ( Y3          ) & 0xFF ] ^ \
                RT1[ ( Y2 >>  8 ) & 0xFF ] ^ \
                RT2[ ( Y1 >> 16 ) & 0xFF ] ^ \
                RT3[ ( Y0 >> 24 ) & 0xFF ]; \
}

```

Definition at line 613 of file aes.c.

Referenced by aes\_crypt\_ecb().

**13.115.1.3 #define GET\_ULONG\_LE(n, b, i)****Value:**

```

{
    (n) = ( (unsigned long) (b)[(i)      ]          ) \
    | ( (unsigned long) (b)[(i) + 1] <<  8 ) \
    | ( (unsigned long) (b)[(i) + 2] << 16 ) \
    | ( (unsigned long) (b)[(i) + 3] << 24 ); \
}

```

Definition at line 40 of file aes.c.

Referenced by aes\_crypt\_ecb(), aes\_setkey\_enc(), and md5\_process().

**13.115.1.4 #define MUL(x, y) ((x && y) ? pow[(log[x]+log[y]) % 255] : 0)**

Definition at line 354 of file aes.c.

Referenced by aes\_gen\_tables().

**13.115.1.5 #define PUT\_ULONG\_LE(n, b, i)****Value:**

```

{
    (b)[(i)      ] = (unsigned char) ( (n)          ); \
    (b)[(i) + 1] = (unsigned char) ( (n) >>  8 ); \
    (b)[(i) + 2] = (unsigned char) ( (n) >> 16 ); \
    (b)[(i) + 3] = (unsigned char) ( (n) >> 24 ); \
}

```

Definition at line 50 of file aes.c.

Referenced by `aes_crypt_ecb()`, and `md5_finish()`.

#### 13.115.1.6 `#define ROTL8(x) ((x << 8) & 0xFFFFFFFF) | (x >> 24)`

Definition at line 352 of file aes.c.

Referenced by `aes_gen_tables()`.

#### 13.115.1.7 `#define XTIME(x) ((x << 1) ^ ((x & 0x80) ? 0x1B : 0x00))`

Definition at line 353 of file aes.c.

Referenced by `aes_gen_tables()`.

### 13.115.2 Function Documentation

#### 13.115.2.1 `void aes_crypt_cbc (aes_context * ctx, int mode, int length, unsigned char iv[16], unsigned char * input, unsigned char * output)`

AES-CBC buffer encryption/decryption.

##### Parameters:

*ctx* AES context  
*mode* AES\_ENCRYPT or AES\_DECRYPT  
*length* length of the input data  
*iv* initialization vector (updated after use)  
*input* buffer holding the input data  
*output* buffer holding the output data

Definition at line 732 of file aes.c.

References `aes_crypt_ecb()`, and `AES_DECRYPT`.

Referenced by `aes_self_test()`, `main()`, `ssl_decrypt_buf()`, and `ssl_encrypt_buf()`.

#### 13.115.2.2 `void aes_crypt_cfb (aes_context * ctx, int mode, int length, int * iv_off, unsigned char iv[16], unsigned char * input, unsigned char * output)`

AES-CFB buffer encryption/decryption.

##### Parameters:

*ctx* AES context  
*mode* AES\_ENCRYPT or AES\_DECRYPT  
*length* length of the input data  
*iv\_off* offset in IV (updated after use)  
*iv* initialization vector (updated after use)  
*input* buffer holding the input data

**output** buffer holding the output data

Definition at line 787 of file aes.c.

References aes\_crypt\_ecb(), AES\_DECRYPT, and AES\_ENCRYPT.

Referenced by aes\_self\_test().

### 13.115.2.3 void aes\_crypt\_ecb (aes\_context \* ctx, int mode, unsigned char input[16], unsigned char output[16])

AES-ECB block encryption/decryption.

#### Parameters:

**ctx** AES context

**mode** AES\_ENCRYPT or AES\_DECRYPT

**input** 16-byte input block

**output** 16-byte output block

Definition at line 639 of file aes.c.

References AES\_DECRYPT, AES\_FROUND, AES\_RROUND, FSb, GET\_ULONG\_LE, aes\_context::nr, PUT\_ULONG\_LE, aes\_context::rk, and RSb.

Referenced by aes\_crypt\_cbc(), aes\_crypt\_cfb(), aes\_en\_de(), aes\_self\_test(), and main().

### 13.115.2.4 static void aes\_gen\_tables (void) [static]

Definition at line 358 of file aes.c.

References FSb, FT0, FT1, FT2, FT3, MUL, RCON, ROTL8, RSb, RT0, RT1, RT2, RT3, and XTIME.

Referenced by aes\_setkey\_enc().

### 13.115.2.5 int aes\_self\_test (int verbose)

Checkup routine.

#### Returns:

0 if successful, or 1 if the test failed

Definition at line 902 of file aes.c.

References aes\_crypt\_cbc(), aes\_crypt\_cfb(), aes\_crypt\_ecb(), AES\_DECRYPT, aes\_setkey\_dec(), aes\_setkey\_enc(), aes\_test\_cbc\_dec, aes\_test\_cbc\_enc, aes\_test\_cfb\_dec, aes\_test\_cfb\_enc, aes\_test\_ecb\_dec, aes\_test\_ecb\_enc, buf, and prv.

Referenced by main().

### 13.115.2.6 void aes\_setkey\_dec (aes\_context \* ctx, unsigned char \* key, int keysize)

AES key schedule (decryption).

**Parameters:**

*ctx* AES context to be initialized  
*key* decryption key  
*keysize* must be 128, 192 or 256

Definition at line 542 of file aes.c.

References `aes_setkey_enc()`, `aes_context::buf`, `FSb`, `aes_context::nr`, `aes_context::rk`, `RT0`, `RT1`, `RT2`, and `RT3`.

Referenced by `aes_en_de()`, `aes_self_test()`, `main()`, and `ssl_derive_keys()`.

### 13.115.2.7 void aes\_setkey\_enc(aes\_context \* ctx, unsigned char \* key, int keysize)

AES key schedule (encryption).

**Parameters:**

*ctx* AES context to be initialized  
*key* encryption key  
*keysize* must be 128, 192 or 256

Definition at line 439 of file aes.c.

References `aes_gen_tables()`, `aes_init_done`, `aes_context::buf`, `FSb`, `GET_ULONG_LE`, `aes_context::nr`, `RCON`, and `aes_context::rk`.

Referenced by `aes_en_de()`, `aes_self_test()`, `aes_setkey_dec()`, `main()`, and `ssl_derive_keys()`.

## 13.115.3 Variable Documentation

### 13.115.3.1 int aes\_init\_done = 0 [static]

Definition at line 356 of file aes.c.

Referenced by `aes_setkey_enc()`.

### 13.115.3.2 const unsigned char aes\_test\_cbc\_dec[3][16] [static]

**Initial value:**

```
{
    { 0xFA, 0xCA, 0x37, 0xE0, 0xB0, 0xC8, 0x53, 0x73,
      0xDF, 0x70, 0x6E, 0x73, 0xF7, 0xC9, 0xAF, 0x86 },
    { 0x5D, 0xF6, 0x78, 0xDD, 0x17, 0xBA, 0x4E, 0x75,
      0xB6, 0x17, 0x68, 0xC6, 0xAD, 0xEF, 0x7C, 0x7B },
    { 0x48, 0x04, 0xE1, 0x81, 0x8F, 0xE6, 0x29, 0x75,
      0x19, 0xA3, 0xE8, 0x8C, 0x57, 0x31, 0x04, 0x13 }
}
```

Definition at line 856 of file aes.c.

Referenced by `aes_self_test()`.

**13.115.3.3   const unsigned char aes\_test\_cbc\_enc[3][16]   [static]****Initial value:**

```
{
    { 0x8A, 0x05, 0xFC, 0x5E, 0x09, 0x5A, 0xF4, 0x84,
      0x8A, 0x08, 0xD3, 0x28, 0xD3, 0x68, 0x8E, 0x3D },
    { 0x7B, 0xD9, 0x66, 0xD5, 0x3A, 0xD8, 0xC1, 0xBB,
      0x85, 0xD2, 0xAD, 0xFA, 0xE8, 0x7B, 0xB1, 0x04 },
    { 0xFE, 0x3C, 0x53, 0x65, 0x3E, 0x2F, 0x45, 0xB5,
      0x6F, 0xCD, 0x88, 0xB2, 0xCC, 0x89, 0x8F, 0xF0 }
}
```

Definition at line 866 of file aes.c.

Referenced by aes\_self\_test().

**13.115.3.4   const unsigned char aes\_test\_cfb\_dec[3][16]   [static]****Initial value:**

```
{
    { 0xBA, 0x75, 0x0C, 0xC9, 0x77, 0xF8, 0xD4, 0xE1,
      0x3E, 0x0F, 0xB5, 0x46, 0x2E, 0xA6, 0x33, 0xF6 },
    { 0xDB, 0x40, 0x4A, 0x98, 0x7B, 0xAA, 0xA3, 0xF3,
      0x92, 0x35, 0xAD, 0x58, 0x09, 0x9B, 0xFF, 0x6E },
    { 0xA8, 0x17, 0x41, 0x0E, 0x76, 0x71, 0x60, 0xE5,
      0xFD, 0x37, 0xC5, 0x43, 0xCC, 0xC8, 0xD6, 0xDA }
}
```

Definition at line 879 of file aes.c.

Referenced by aes\_self\_test().

**13.115.3.5   const unsigned char aes\_test\_cfb\_enc[3][16]   [static]****Initial value:**

```
{
    { 0x45, 0x62, 0xC5, 0xA1, 0xF9, 0x10, 0x8F, 0xE0,
      0x87, 0x24, 0x25, 0x68, 0xB5, 0x12, 0xF3, 0x8B },
    { 0xB8, 0xD4, 0xD5, 0x09, 0xF5, 0xEE, 0x08, 0x38,
      0x48, 0x9B, 0x9D, 0xAD, 0x11, 0xB4, 0x2E, 0xD2 },
    { 0xE9, 0x10, 0x80, 0xDA, 0xEE, 0x2D, 0x81, 0xD9,
      0x41, 0x78, 0x91, 0xD5, 0x98, 0x78, 0xE1, 0xFA }
}
```

Definition at line 889 of file aes.c.

Referenced by aes\_self\_test().

**13.115.3.6   const unsigned char aes\_test\_ecb\_dec[3][16]   [static]****Initial value:**



```
{
    { 0x44, 0x41, 0x6A, 0xC2, 0xD1, 0xF5, 0x3C, 0x58,
      0x33, 0x03, 0x91, 0x7E, 0x6B, 0xE9, 0xEB, 0xE0 },
    { 0x48, 0xE3, 0x1E, 0x9E, 0x25, 0x67, 0x18, 0xF2,
      0x92, 0x29, 0x31, 0x9C, 0x19, 0xF1, 0x5B, 0xA4 },
    { 0x05, 0x8C, 0xCF, 0xFD, 0xBB, 0xCB, 0x38, 0x2D,
      0x1F, 0x6F, 0x56, 0x58, 0x5D, 0x8A, 0x4A, 0xDE }
}
```

Definition at line 836 of file aes.c.

Referenced by aes\_self\_test().

### 13.115.3.7 const unsigned char aes\_test\_ecb\_enc[3][16] [static]

Initial value:

```
{
    { 0xC3, 0x4C, 0x05, 0x2C, 0xC0, 0xDA, 0x8D, 0x73,
      0x45, 0x1A, 0xFE, 0x5F, 0x03, 0xBE, 0x29, 0x7F },
    { 0xF3, 0xF6, 0x75, 0x2A, 0xE8, 0xD7, 0x83, 0x11,
      0x38, 0xF0, 0x41, 0x56, 0x06, 0x31, 0xB1, 0x14 },
    { 0x8B, 0x79, 0xEE, 0xCC, 0x93, 0xA0, 0xEE, 0x5D,
      0xFF, 0x30, 0xB4, 0xEA, 0x21, 0x63, 0x6D, 0xA4 }
}
```

Definition at line 846 of file aes.c.

Referenced by aes\_self\_test().

### 13.115.3.8 unsigned char FSb[256] [static]

Definition at line 329 of file aes.c.

Referenced by aes\_crypt\_ecb(), aes\_gen\_tables(), aes\_setkey\_dec(), and aes\_setkey\_enc().

### 13.115.3.9 unsigned long FT0[256] [static]

Definition at line 330 of file aes.c.

Referenced by aes\_gen\_tables().

### 13.115.3.10 unsigned long FT1[256] [static]

Definition at line 331 of file aes.c.

Referenced by aes\_gen\_tables().

### 13.115.3.11 unsigned long FT2[256] [static]

Definition at line 332 of file aes.c.

Referenced by aes\_gen\_tables().

**13.115.3.12 unsigned long FT3[256] [static]**

Definition at line 333 of file aes.c.

Referenced by aes\_gen\_tables().

**13.115.3.13 unsigned long RCON[10] [static]**

Definition at line 347 of file aes.c.

Referenced by aes\_gen\_tables(), and aes\_setkey\_enc().

**13.115.3.14 unsigned char RSb[256] [static]**

Definition at line 338 of file aes.c.

Referenced by aes\_crypt\_ecb(), and aes\_gen\_tables().

**13.115.3.15 unsigned long RT0[256] [static]**

Definition at line 339 of file aes.c.

Referenced by aes\_gen\_tables(), and aes\_setkey\_dec().

**13.115.3.16 unsigned long RT1[256] [static]**

Definition at line 340 of file aes.c.

Referenced by aes\_gen\_tables(), and aes\_setkey\_dec().

**13.115.3.17 unsigned long RT2[256] [static]**

Definition at line 341 of file aes.c.

Referenced by aes\_gen\_tables(), and aes\_setkey\_dec().

**13.115.3.18 unsigned long RT3[256] [static]**

Definition at line 342 of file aes.c.

Referenced by aes\_gen\_tables(), and aes\_setkey\_dec().

## 13.116 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/library/arc4.c File Reference

```
#include "xyssl/config.h"
#include "xyssl/arc4.h"
#include <string.h>
#include <stdio.h>
```

### Functions

- void [arc4\\_setup](#) ([arc4\\_context](#) \*ctx, unsigned char \*key, [int](#) keylen)  
*ARC4 key schedule.*
- void [arc4\\_crypt](#) ([arc4\\_context](#) \*ctx, unsigned char \*buf, [int](#) buflen)  
*ARC4 cipher function.*
- [int](#) [arc4\\_self\\_test](#) ([int](#) verbose)

### Variables

- static const unsigned char [arc4\\_test\\_key](#) [3][8]
- static const unsigned char [arc4\\_test\\_pt](#) [3][8]
- static const unsigned char [arc4\\_test\\_ct](#) [3][8]

### 13.116.1 Function Documentation

#### 13.116.1.1 void [arc4\\_crypt](#) ([arc4\\_context](#) \*ctx, unsigned char \*buf, [int](#) buflen)

ARC4 cipher function.

##### Parameters:

*ctx* ARC4 context  
*buf* buffer to be processed  
*buflen* amount of data in buf

Definition at line 63 of file arc4.c.

References [arc4\\_context::m](#), [arc4\\_context::x](#), and [arc4\\_context::y](#).

Referenced by [arc4\\_self\\_test\(\)](#), [main\(\)](#), [ssl\\_decrypt\\_buf\(\)](#), and [ssl\\_encrypt\\_buf\(\)](#).

#### 13.116.1.2 [int](#) [arc4\\_self\\_test](#) ([int](#) verbose)

Definition at line 122 of file arc4.c.

References [arc4\\_crypt\(\)](#), [arc4\\_setup\(\)](#), [arc4\\_test\\_ct](#), [arc4\\_test\\_key](#), [arc4\\_test\\_pt](#), and [buf](#).

Referenced by [main\(\)](#).

### 13.116.1.3 void arc4\_setup (arc4\_context \* *ctx*, unsigned char \* *key*, int *keylen*)

ARC4 key schedule.

#### Parameters:

*ctx* ARC4 context to be initialized

*key* the secret key

*keylen* length of the key

Definition at line 35 of file arc4.c.

References arc4\_context::m, arc4\_context::x, and arc4\_context::y.

Referenced by arc4\_self\_test(), main(), and ssl\_derive\_keys().

## 13.116.2 Variable Documentation

### 13.116.2.1 const unsigned char arc4\_test\_ct[3][8] [static]

#### Initial value:

```
{
    { 0x75, 0xB7, 0x87, 0x80, 0x99, 0xE0, 0xC5, 0x96 },
    { 0x74, 0x94, 0xC2, 0xE7, 0x10, 0x4B, 0x08, 0x79 },
    { 0xDE, 0x18, 0x89, 0x41, 0xA3, 0x37, 0x5D, 0x3A }
}
```

Definition at line 112 of file arc4.c.

Referenced by arc4\_self\_test().

### 13.116.2.2 const unsigned char arc4\_test\_key[3][8] [static]

#### Initial value:

```
{
    { 0x01, 0x23, 0x45, 0x67, 0x89, 0xAB, 0xCD, 0xEF },
    { 0x01, 0x23, 0x45, 0x67, 0x89, 0xAB, 0xCD, 0xEF },
    { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 }
}
```

Definition at line 98 of file arc4.c.

Referenced by arc4\_self\_test().

### 13.116.2.3 const unsigned char arc4\_test\_pt[3][8] [static]

#### Initial value:

```
{
    { 0x01, 0x23, 0x45, 0x67, 0x89, 0xAB, 0xCD, 0xEF },
    { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 }
}
```

Definition at line 105 of file arc4.c.

Referenced by arc4\_self\_test().

## 13.117 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/library/base64.c File Reference

```
#include "xyssl/config.h"
#include "xyssl/base64.h"
#include <string.h>
#include <stdio.h>
```

### Functions

- [int base64\\_encode](#) (unsigned char \*dst, [int](#) \*dlen, unsigned char \*src, [int](#) slen)  
*Encode a buffer into base64 format.*
- [int base64\\_decode](#) (unsigned char \*dst, [int](#) \*dlen, unsigned char \*src, [int](#) slen)  
*Decode a base64-formatted buffer.*
- [int base64\\_self\\_test](#) ([int](#) verbose)  
*Checkup routine.*

### Variables

- static const unsigned char [base64\\_enc\\_map](#) [64]
- static const unsigned char [base64\\_dec\\_map](#) [128]
- static const unsigned char [base64\\_test\\_dec](#) [64]
- static const unsigned char [base64\\_test\\_enc](#) [ ]

### 13.117.1 Function Documentation

#### 13.117.1.1 [int base64\\_decode](#) (unsigned char \* dst, [int](#) \* dlen, unsigned char \* src, [int](#) slen)

Decode a base64-formatted buffer.

##### Parameters:

*dst* destination buffer  
*dlen* size of the buffer  
*src* source buffer  
*slen* amount of data to be decoded

##### Returns:

0 if successful, XYSSL\_ERR\_BASE64\_BUFFER\_TOO\_SMALL, or XYSSL\_ERR\_BASE64\_INVALID\_DATA if the input data is not correct. \*dlen is always updated to reflect the amount of data that has (or would have) been written.

**Note:**

Call this function with `*dlen = 0` to obtain the required buffer size in `*dlen`

Definition at line 121 of file `base64.c`.

References `base64_dec_map`, `XYSSL_ERR_BASE64_BUFFER_TOO_SMALL`, and `XYSSL_ERR_BASE64_INVALID_CHARACTER`.

Referenced by `base64_self_test()`, `x509parse_crt()`, and `x509parse_key()`.

**13.117.1.2 int base64\_encode (unsigned char \* *dst*, int \* *dlen*, unsigned char \* *src*, int *slen*)**

Encode a buffer into base64 format.

**Parameters:**

*dst* destination buffer

*dlen* size of the buffer

*src* source buffer

*slen* amount of data to be encoded

**Returns:**

0 if successful, or `XYSSL_ERR_BASE64_BUFFER_TOO_SMALL`. `*dlen` is always updated to reflect the amount of data that has (or would have) been written.

**Note:**

Call this function with `*dlen = 0` to obtain the required buffer size in `*dlen`

Definition at line 58 of file `base64.c`.

References `base64_enc_map`, and `XYSSL_ERR_BASE64_BUFFER_TOO_SMALL`.

Referenced by `base64_self_test()`.

**13.117.1.3 int base64\_self\_test (int *verbose*)**

Checkup routine.

**Returns:**

0 if successful, or 1 if the test failed

Definition at line 206 of file `base64.c`.

References `base64_decode()`, `base64_encode()`, `base64_test_dec`, and `base64_test_enc`.

Referenced by `main()`.

**13.117.2 Variable Documentation****13.117.2.1 const unsigned char base64\_dec\_map[128] [static]****Initial value:**

```
{
    127, 127, 127, 127, 127, 127, 127, 127, 127, 127,
    127, 127, 127, 127, 127, 127, 127, 127, 127, 127,
    127, 127, 127, 127, 127, 127, 127, 127, 127, 127,
    127, 127, 127, 127, 127, 127, 127, 127, 127, 127,
    127, 127, 127, 62, 127, 127, 127, 63, 52, 53,
    54, 55, 56, 57, 58, 59, 60, 61, 127, 127,
    127, 64, 127, 127, 127, 0, 1, 2, 3, 4,
    5, 6, 7, 8, 9, 10, 11, 12, 13, 14,
    15, 16, 17, 18, 19, 20, 21, 22, 23, 24,
    25, 127, 127, 127, 127, 127, 127, 127, 26, 27, 28,
    29, 30, 31, 32, 33, 34, 35, 36, 37, 38,
    39, 40, 41, 42, 43, 44, 45, 46, 47, 48,
    49, 50, 51, 127, 127, 127, 127, 127
}
```

Definition at line 38 of file base64.c.

Referenced by base64\_decode().

### 13.117.2.2 `const unsigned char base64_enc_map[64] [static]`

**Initial value:**

```
{
    'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J',
    'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T',
    'U', 'V', 'W', 'X', 'Y', 'Z', 'a', 'b', 'c', 'd',
    'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n',
    'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x',
    'y', 'z', '0', '1', '2', '3', '4', '5', '6', '7',
    '8', '9', '+', '/'
}
```

Definition at line 27 of file base64.c.

Referenced by base64\_encode().

### 13.117.2.3 `const unsigned char base64_test_dec[64] [static]`

**Initial value:**

```
{
    0x24, 0x48, 0x6E, 0x56, 0x87, 0x62, 0x5A, 0xBD,
    0xBF, 0x17, 0xD9, 0xA2, 0xC4, 0x17, 0x1A, 0x01,
    0x94, 0xED, 0x8F, 0x1E, 0x11, 0xB3, 0xD7, 0x09,
    0x0C, 0xB6, 0xE9, 0x10, 0x6F, 0x22, 0xEE, 0x13,
    0xCA, 0xB3, 0x07, 0x05, 0x76, 0xC9, 0xFA, 0x31,
    0x6C, 0x08, 0x34, 0xFF, 0x8D, 0xC2, 0x6C, 0x38,
    0x00, 0x43, 0xE9, 0x54, 0x97, 0xAF, 0x50, 0x4B,
    0xD1, 0x41, 0xBA, 0x95, 0x31, 0x5A, 0x0B, 0x97
}
```

Definition at line 187 of file base64.c.

Referenced by base64\_self\_test().



#### 13.117.2.4 `const unsigned char base64_test_enc[]` `[static]`

**Initial value:**

```
"JEhuVodiWr2/F9mixBcaAZTtjx4Rs9cJDLbpEG8i7hPK"  
"swcFdsn6MWwINP+Nwmw4AEPpVJevUEvRQbqVMVoLlw=="
```

Definition at line 199 of file base64.c.

Referenced by base64\_self\_test().

## 13.118 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/library/bignum.c File Reference

```
#include "xyssl/config.h"
#include "xyssl/bignum.h"
#include "xyssl/bn_mul.h"
#include <string.h>
#include <stdlib.h>
#include <stdarg.h>
```

### Defines

- `#define ciL ((int) sizeof(t_int))`
- `#define biL (ciL << 3)`
- `#define biH (ciL << 2)`
- `#define BITS_TO_LIMBS(i) (((i) + biL - 1) / biL)`
- `#define CHARS_TO_LIMBS(i) (((i) + ciL - 1) / ciL)`

### Functions

- `void mpi_init (mpi *X,...)`  
*Initialize one or more mpi.*
- `void mpi_free (mpi *X,...)`  
*Unallocate one or more mpi.*
- `int mpi_grow (mpi *X, int nblimbs)`  
*Enlarge to the specified number of limbs.*
- `int mpi_copy (mpi *X, mpi *Y)`  
*Copy the contents of Y into X.*
- `void mpi_swap (mpi *X, mpi *Y)`  
*Swap the contents of X and Y.*
- `int mpi_lset (mpi *X, int z)`  
*Set value from integer.*
- `int mpi_lsb (mpi *X)`  
*Return the number of least significant bits.*
- `int mpi_msb (mpi *X)`  
*Return the number of most significant bits.*
- `int mpi_size (mpi *X)`

*Return the total size in bytes.*

- static `int mpi_get_digit (t_int *d, int radix, char c)`
- `int mpi_read_string (mpi *X, int radix, char *s)`

*Import from an ASCII string.*

- static `int mpi_write_hlp (mpi *X, int radix, char **p)`
- `int mpi_write_string (mpi *X, int radix, char *s, int *slen)`

*Export into an ASCII string.*

- `int mpi_read_file (mpi *X, int radix, FILE *fin)`

*Read X from an opened file.*

- `int mpi_read_mystring (mpi *X, int radix, char *s)`
- `int mpi_write_file (char *p, mpi *X, int radix, FILE *fout)`

*Write X into an opened file, or stdout.*

- `int mpi_read_binary (mpi *X, unsigned char *buf, int buflen)`

*Import X from unsigned binary data, big endian.*

- `int mpi_write_binary (mpi *X, unsigned char *buf, int buflen)`

*Export X into unsigned binary data, big endian.*

- `int mpi_shift_l (mpi *X, int count)`

*Left-shift:  $X \ll = \text{count}$ .*

- `int mpi_shift_r (mpi *X, int count)`

*Right-shift:  $X \gg = \text{count}$ .*

- `int mpi_cmp_abs (mpi *X, mpi *Y)`

*Compare unsigned values.*

- `int mpi_cmp_mpi (mpi *X, mpi *Y)`

*Compare signed values.*

- `int mpi_cmp_int (mpi *X, int z)`

*Compare signed values.*

- `int mpi_add_abs (mpi *X, mpi *A, mpi *B)`

*Unsigned addition:  $X = |A| + |B|$ .*

- static void `mpi_sub_hlp (int n, t_int *s, t_int *d)`

- `int mpi_sub_abs (mpi *X, mpi *A, mpi *B)`

*Unsigned subtraction:  $X = |A| - |B|$ .*

- `int mpi_add_mpi (mpi *X, mpi *A, mpi *B)`

*Signed addition:  $X = A + B$ .*

- `int mpi_sub_mpi (mpi *X, mpi *A, mpi *B)`

*Signed subtraction:  $X = A - B$ .*

- `int mpi_add_int (mpi *X, mpi *A, int b)`  
Signed addition:  $X = A + b$ .
- `int mpi_sub_int (mpi *X, mpi *A, int b)`  
Signed subtraction:  $X = A - b$ .
- `static void mpi_mul_hlp (int i, t_int *s, t_int *d, t_int b)`
- `int mpi_mul_mpi (mpi *X, mpi *A, mpi *B)`  
Baseline multiplication:  $X = A * B$ .
- `int mpi_mul_int (mpi *X, mpi *A, t_int b)`  
Baseline multiplication:  $X = A * b$ .
- `int mpi_div_mpi (mpi *Q, mpi *R, mpi *A, mpi *B)`  
Division by *mpi*:  $A = Q * B + R$ .
- `int mpi_div_int (mpi *Q, mpi *R, mpi *A, int b)`  
Division by *int*:  $A = Q * b + R$ .
- `int mpi_mod_mpi (mpi *R, mpi *A, mpi *B)`  
Modulo:  $R = A \bmod B$ .
- `int mpi_mod_int (t_int *r, mpi *A, int b)`  
Modulo:  $r = A \bmod b$ .
- `static void mpi_montg_init (t_int *mm, mpi *N)`
- `static void mpi_montmul (mpi *A, mpi *B, mpi *N, t_int mm, mpi *T)`
- `static void mpi_montred (mpi *A, mpi *N, t_int mm, mpi *T)`
- `int mpi_exp_mod (mpi *X, mpi *A, mpi *E, mpi *N, mpi *_RR)`  
Sliding-window exponentiation:  $X = A^E \bmod N$ .
- `int mpi_gcd (mpi *G, mpi *A, mpi *B)`  
Greatest common divisor:  $G = \gcd(A, B)$ .
- `int mpi_inv_mod (mpi *X, mpi *A, mpi *N)`  
Modular inverse:  $X = A^{-1} \bmod N$ .
- `int mpi_is_prime (mpi *X, int(*f_rng)(void *), void *p_rng)`  
Miller-Rabin primality test.
- `int mpi_gen_prime (mpi *X, int nbits, int dh_flag, int(*f_rng)(void *), void *p_rng)`  
Prime number generation.
- `int mpi_self_test (int verbose)`  
Checkup routine.

## Variables

- static const `int small_prime` [ ]

### 13.118.1 Define Documentation

#### 13.118.1.1 `#define biH (ciL << 2)`

Definition at line 41 of file `bignum.c`.

Referenced by `mpi_div_mpi()`, and `mpi_mod_int()`.

#### 13.118.1.2 `#define biL (ciL << 3)`

Definition at line 40 of file `bignum.c`.

Referenced by `mpi_div_mpi()`, `mpi_exp_mod()`, `mpi_lsb()`, `mpi_montg_init()`, `mpi_msb()`, `mpi_shift_l()`, and `mpi_shift_r()`.

#### 13.118.1.3 `#define BITS_TO_LIMBS(i) (((i) + biL - 1) / biL)`

Definition at line 46 of file `bignum.c`.

Referenced by `mpi_gen_prime()`, `mpi_read_string()`, and `mpi_shift_l()`.

#### 13.118.1.4 `#define CHARS_TO_LIMBS(i) (((i) + ciL - 1) / ciL)`

Definition at line 47 of file `bignum.c`.

Referenced by `mpi_read_binary()`.

#### 13.118.1.5 `#define ciL ((int) sizeof(t_int))`

Definition at line 39 of file `bignum.c`.

Referenced by `mpi_copy()`, `mpi_free()`, `mpi_gen_prime()`, `mpi_grow()`, `mpi_is_prime()`, `mpi_lset()`, `mpi_montmul()`, `mpi_read_binary()`, `mpi_read_string()`, `mpi_write_binary()`, and `mpi_write_string()`.

### 13.118.2 Function Documentation

#### 13.118.2.1 `int mpi_add_abs (mpi * X, mpi * A, mpi * B)`

Unsigned addition:  $X = |A| + |B|$ .

##### Returns:

0 if successful, 1 if memory allocation failed

Definition at line 694 of file `bignum.c`.

References `MPI_CHK`, `mpi_copy()`, `mpi_grow()`, `mpi::n`, and `mpi::p`.

Referenced by `mpi_add_mpi()`, and `mpi_sub_mpi()`.

**13.118.2.2 int mpi\_add\_int (mpi \* X, mpi \* A, int b)**

Signed addition:  $X = A + b$ .

**Returns:**

0 if successful, 1 if memory allocation failed

Definition at line 860 of file bignum.c.

References mpi\_add\_mpi(), mpi::n, mpi::p, and mpi::s.

Referenced by mpi\_gen\_prime(), and mpi\_read\_string().

**13.118.2.3 int mpi\_add\_mpi (mpi \* X, mpi \* A, mpi \* B)**

Signed addition:  $X = A + B$ .

**Returns:**

0 if successful, 1 if memory allocation failed

Definition at line 798 of file bignum.c.

References mpi\_add\_abs(), MPI\_CHK, mpi\_cmp\_abs(), mpi\_sub\_abs(), and mpi::s.

Referenced by mpi\_add\_int(), mpi\_div\_mpi(), mpi\_inv\_mod(), mpi\_mod\_mpi(), and rsa\_private().

**13.118.2.4 int mpi\_cmp\_abs (mpi \* X, mpi \* Y)**

Compare unsigned values.

**Returns:**

1 if  $|X|$  is greater than  $|Y|$ , -1 if  $|X|$  is lesser than  $|Y|$  or 0 if  $|X|$  is equal to  $|Y|$

Definition at line 615 of file bignum.c.

References mpi::n, and mpi::p.

Referenced by mpi\_add\_mpi(), mpi\_div\_mpi(), mpi\_montmul(), mpi\_sub\_abs(), and mpi\_sub\_mpi().

**13.118.2.5 int mpi\_cmp\_int (mpi \* X, int z)**

Compare signed values.

**Returns:**

1 if X is greater than z, -1 if X is lesser than z or 0 if X is equal to z

Definition at line 678 of file bignum.c.

References mpi\_cmp\_mpi(), mpi::n, mpi::p, and mpi::s.

Referenced by mpi\_div\_mpi(), mpi\_exp\_mod(), mpi\_gcd(), mpi\_inv\_mod(), mpi\_is\_prime(), mpi\_mod\_mpi(), mpi\_write\_hlp(), rsa\_check\_privkey(), and rsa\_gen\_key().

### 13.118.2.6 `int mpi_cmp_mpi (mpi * X, mpi * Y)`

Compare signed values.

#### Returns:

1 if X is greater than Y, -1 if X is lesser than Y or 0 if X is equal to Y

Definition at line 645 of file bignum.c.

References `mpi::n`, `mpi::p`, and `mpi::s`.

Referenced by `dhm_make_params()`, `dhm_make_public()`, `mpi_cmp_int()`, `mpi_div_mpi()`, `mpi_exp_mod()`, `mpi_gcd()`, `mpi_inv_mod()`, `mpi_is_prime()`, `mpi_mod_mpi()`, `mpi_self_test()`, `rsa_check_privkey()`, `rsa_gen_key()`, `rsa_private()`, and `rsa_public()`.

### 13.118.2.7 `int mpi_copy (mpi * X, mpi * Y)`

Copy the contents of Y into X.

#### Returns:

0 if successful, 1 if memory allocation failed

Definition at line 128 of file bignum.c.

References `ciL`, `MPI_CHK`, `mpi_grow()`, `mpi::n`, `mpi::p`, and `mpi::s`.

Referenced by `mpi_add_abs()`, `mpi_div_mpi()`, `mpi_exp_mod()`, `mpi_gcd()`, `mpi_inv_mod()`, `mpi_is_prime()`, `mpi_mul_mpi()`, `mpi_sub_abs()`, and `mpi_write_string()`.

### 13.118.2.8 `int mpi_div_int (mpi * Q, mpi * R, mpi * A, int b)`

Division by int:  $A = Q * b + R$ .

#### Returns:

0 if successful, 1 if memory allocation failed, `XYSSL_ERR_MPI_DIVISION_BY_ZERO` if  $b == 0$

#### Note:

Either Q or R can be NULL.

Definition at line 1173 of file bignum.c.

References `mpi_div_mpi()`, `mpi::n`, `mpi::p`, and `mpi::s`.

Referenced by `main()`, and `mpi_write_hlp()`.

### 13.118.2.9 `int mpi_div_mpi (mpi * Q, mpi * R, mpi * A, mpi * B)`

Division by `mpi`:  $A = Q * B + R$ .

#### Returns:

0 if successful, 1 if memory allocation failed, `XYSSL_ERR_MPI_DIVISION_BY_ZERO` if  $B == 0$

**Note:**

Either Q or R can be NULL.

Definition at line 1008 of file bignum.c.

References biH, biL, mpi\_add\_mpi(), MPI\_CHK, mpi\_cmp\_abs(), mpi\_cmp\_int(), mpi\_cmp\_mpi(), mpi\_copy(), mpi\_free(), mpi\_grow(), mpi\_init(), mpi\_lset(), mpi\_msb(), mpi\_mul\_int(), mpi\_shift\_l(), mpi\_shift\_r(), mpi\_sub\_mpi(), mpi::n, mpi::p, mpi::s, and XYSSL\_ERR\_MPI\_DIVISION\_BY\_ZERO.

Referenced by mpi\_div\_int(), mpi\_mod\_mpi(), and mpi\_self\_test().

**13.118.2.10 int mpi\_exp\_mod (mpi \* X, mpi \* A, mpi \* E, mpi \* N, mpi \* \_RR)**

Sliding-window exponentiation:  $X = A^E \bmod N$ .

**Returns:**

0 if successful, 1 if memory allocation failed, XYSSL\_ERR\_MPI\_BAD\_INPUT\_DATA if N is negative or even

**Note:**

\_RR is used to avoid re-computing  $R \cdot R \bmod N$  across multiple calls, which speeds up things a bit. It can be set to NULL if the extra performance is unneeded.

Definition at line 1328 of file bignum.c.

References biL, MPI\_CHK, mpi\_cmp\_int(), mpi\_cmp\_mpi(), mpi\_copy(), mpi\_free(), mpi\_grow(), mpi\_init(), mpi\_lset(), mpi\_mod\_mpi(), mpi\_montg\_init(), mpi\_montmul(), mpi\_montred(), mpi\_msb(), mpi\_shift\_l(), mpi::n, mpi::p, and XYSSL\_ERR\_MPI\_BAD\_INPUT\_DATA.

Referenced by dhm\_calc\_secret(), dhm\_make\_params(), dhm\_make\_public(), main(), mpi\_is\_prime(), mpi\_self\_test(), rsa\_private(), and rsa\_public().

**13.118.2.11 void mpi\_free (mpi \* X, ...)**

Unallocate one or more [mpi](#).

Definition at line 73 of file bignum.c.

References ciL, mpi::n, mpi::p, and mpi::s.

Referenced by dhm\_free(), main(), mpi\_div\_mpi(), mpi\_exp\_mod(), mpi\_gcd(), mpi\_gen\_prime(), mpi\_inv\_mod(), mpi\_is\_prime(), mpi\_mul\_mpi(), mpi\_read\_string(), mpi\_self\_test(), mpi\_sub\_abs(), mpi\_write\_string(), rsa\_check\_privkey(), rsa\_free(), rsa\_gen\_key(), rsa\_private(), and rsa\_public().

**13.118.2.12 int mpi\_gcd (mpi \* G, mpi \* A, mpi \* B)**

Greatest common divisor:  $G = \gcd(A, B)$ .

**Returns:**

0 if successful, 1 if memory allocation failed

Definition at line 1507 of file bignum.c.



References MPI\_CHK, mpi\_cmp\_int(), mpi\_cmp\_mpi(), mpi\_copy(), mpi\_free(), mpi\_init(), mpi\_lset(), mpi\_mul\_mpi(), mpi\_shift\_r(), mpi\_sub\_abs(), mpi::p, and mpi::s.

Referenced by mpi\_inv\_mod(), rsa\_check\_privkey(), and rsa\_gen\_key().

### 13.118.2.13 int mpi\_gen\_prime (mpi \* X, int nbits, int dh\_flag, int(\*) (void \*) f\_rng, void \* p\_rng)

Prime number generation.

#### Parameters:

*X* destination [mpi](#)  
*nbits* required size of X in bits  
*dh\_flag* if 1, then (X-1)/2 will be prime too  
*f\_rng* RNG function  
*p\_rng* RNG parameter

#### Returns:

0 if successful (probably prime), 1 if memory allocation failed, XYSSL\_ERR\_MPI\_BAD\_INPUT\_DATA if nbits is < 3

Definition at line 1778 of file bignum.c.

References BITS\_TO\_LIMBS, ciL, mpi\_add\_int(), MPI\_CHK, mpi\_free(), mpi\_grow(), mpi\_init(), mpi\_is\_prime(), mpi\_lset(), mpi\_msb(), mpi\_shift\_l(), mpi\_shift\_r(), mpi\_sub\_int(), mpi::p, XYSSL\_ERR\_MPI\_BAD\_INPUT\_DATA, and XYSSL\_ERR\_MPI\_NOT\_ACCEPTABLE.

Referenced by main(), and rsa\_gen\_key().

### 13.118.2.14 static int mpi\_get\_digit (t\_int \* d, int radix, char c) [static]

Definition at line 226 of file bignum.c.

References XYSSL\_ERR\_MPI\_INVALID\_CHARACTER.

Referenced by mpi\_read\_file(), mpi\_read\_mystring(), and mpi\_read\_string().

### 13.118.2.15 int mpi\_grow (mpi \* X, int nlimbs)

Enlarge to the specified number of limbs.

#### Returns:

0 if successful, 1 if memory allocation failed

Definition at line 100 of file bignum.c.

References ciL, mpi::n, and mpi::p.

Referenced by dhm\_make\_params(), dhm\_make\_public(), mpi\_add\_abs(), mpi\_copy(), mpi\_div\_mpi(), mpi\_exp\_mod(), mpi\_gen\_prime(), mpi\_is\_prime(), mpi\_lset(), mpi\_mul\_mpi(), mpi\_read\_binary(), mpi\_read\_string(), and mpi\_shift\_l().

**13.118.2.16 void mpi\_init (mpi \* X, ...)**

Initialize one or more [mpi](#).

Definition at line 52 of file bignum.c.

References [mpi::n](#), [mpi::p](#), and [mpi::s](#).

Referenced by [main\(\)](#), [mpi\\_div\\_mpi\(\)](#), [mpi\\_exp\\_mod\(\)](#), [mpi\\_gcd\(\)](#), [mpi\\_gen\\_prime\(\)](#), [mpi\\_inv\\_mod\(\)](#), [mpi\\_is\\_prime\(\)](#), [mpi\\_mul\\_mpi\(\)](#), [mpi\\_read\\_string\(\)](#), [mpi\\_self\\_test\(\)](#), [mpi\\_sub\\_abs\(\)](#), [mpi\\_write\\_string\(\)](#), [rsa\\_check\\_privkey\(\)](#), [rsa\\_gen\\_key\(\)](#), [rsa\\_private\(\)](#), and [rsa\\_public\(\)](#).

**13.118.2.17 int mpi\_inv\_mod (mpi \* X, mpi \* A, mpi \* N)**

Modular inverse:  $X = A^{-1} \bmod N$ .

**Returns:**

0 if successful, 1 if memory allocation failed, [XYSSL\\_ERR\\_MPI\\_BAD\\_INPUT\\_DATA](#) if N is negative or nil [XYSSL\\_ERR\\_MPI\\_NOT\\_ACCEPTABLE](#) if A has no inverse mod N

Definition at line 1549 of file bignum.c.

References [mpi\\_add\\_mpi\(\)](#), [MPI\\_CHK](#), [mpi\\_cmp\\_int\(\)](#), [mpi\\_cmp\\_mpi\(\)](#), [mpi\\_copy\(\)](#), [mpi\\_free\(\)](#), [mpi\\_gcd\(\)](#), [mpi\\_init\(\)](#), [mpi\\_lset\(\)](#), [mpi\\_mod\\_mpi\(\)](#), [mpi\\_shift\\_r\(\)](#), [mpi\\_sub\\_mpi\(\)](#), [mpi::p](#), [XYSSL\\_ERR\\_MPI\\_BAD\\_INPUT\\_DATA](#), and [XYSSL\\_ERR\\_MPI\\_NOT\\_ACCEPTABLE](#).

Referenced by [main\(\)](#), [mpi\\_self\\_test\(\)](#), and [rsa\\_gen\\_key\(\)](#).

**13.118.2.18 int mpi\_is\_prime (mpi \* X, int(\*) (void \*) f\_rng, void \* p\_rng)**

Miller-Rabin primality test.

**Returns:**

0 if successful (probably prime), 1 if memory allocation failed, [XYSSL\\_ERR\\_MPI\\_NOT\\_ACCEPTABLE](#) if X is not prime

Definition at line 1667 of file bignum.c.

References [ciL](#), [MPI\\_CHK](#), [mpi\\_cmp\\_int\(\)](#), [mpi\\_cmp\\_mpi\(\)](#), [mpi\\_copy\(\)](#), [mpi\\_exp\\_mod\(\)](#), [mpi\\_free\(\)](#), [mpi\\_grow\(\)](#), [mpi\\_init\(\)](#), [mpi\\_lsb\(\)](#), [mpi\\_mod\\_int\(\)](#), [mpi\\_mod\\_mpi\(\)](#), [mpi\\_msb\(\)](#), [mpi\\_mul\\_mpi\(\)](#), [mpi\\_shift\\_r\(\)](#), [mpi\\_sub\\_int\(\)](#), [mpi::n](#), [mpi::p](#), [R](#), [mpi::s](#), [small\\_prime](#), and [XYSSL\\_ERR\\_MPI\\_NOT\\_ACCEPTABLE](#).

Referenced by [main\(\)](#), and [mpi\\_gen\\_prime\(\)](#).

**13.118.2.19 int mpi\_lsb (mpi \* X)**

Return the number of least significant bits.

Definition at line 185 of file bignum.c.

References [biL](#), [int](#), [mpi::n](#), and [mpi::p](#).

Referenced by [mpi\\_is\\_prime\(\)](#).

### 13.118.2.20 `int mpi_lset (mpi * X, int z)`

Set value from integer.

#### Returns:

0 if successful, 1 if memory allocation failed

Definition at line 167 of file bignum.c.

References `ciL`, `MPI_CHK`, `mpi_grow()`, `mpi::n`, `mpi::p`, and `mpi::s`.

Referenced by `dhm_make_params()`, `dhm_make_public()`, `mpi_div_mpi()`, `mpi_exp_mod()`, `mpi_gcd()`, `mpi_gen_prime()`, `mpi_inv_mod()`, `mpi_mul_mpi()`, `mpi_read_binary()`, `mpi_read_string()`, and `rsa_gen_key()`.

### 13.118.2.21 `int mpi_mod_int (t_int * r, mpi * A, int b)`

Modulo:  $r = A \bmod b$ .

#### Returns:

0 if successful, 1 if memory allocation failed, `XYSSL_ERR_MPI_DIVISION_BY_ZERO` if  $b == 0$

Definition at line 1209 of file bignum.c.

References `biH`, `mpi::n`, `mpi::p`, and `XYSSL_ERR_MPI_DIVISION_BY_ZERO`.

Referenced by `mpi_is_prime()`, and `mpi_write_hlp()`.

### 13.118.2.22 `int mpi_mod_mpi (mpi * R, mpi * A, mpi * B)`

Modulo:  $R = A \bmod B$ .

#### Returns:

0 if successful, 1 if memory allocation failed, `XYSSL_ERR_MPI_DIVISION_BY_ZERO` if  $B == 0$

Definition at line 1189 of file bignum.c.

References `mpi_add_mpi()`, `MPI_CHK`, `mpi_cmp_int()`, `mpi_cmp_mpi()`, `mpi_div_mpi()`, and `mpi_sub_mpi()`.

Referenced by `mpi_exp_mod()`, `mpi_inv_mod()`, `mpi_is_prime()`, `rsa_check_privkey()`, `rsa_gen_key()`, and `rsa_private()`.

### 13.118.2.23 `static void mpi_montg_init (t_int * mm, mpi * N) [static]`

Definition at line 1259 of file bignum.c.

References `biL`, and `mpi::p`.

Referenced by `mpi_exp_mod()`.

### 13.118.2.24 `static void mpi_montmul (mpi * A, mpi * B, mpi * N, t_int mm, mpi * T) [static]`

Definition at line 1277 of file bignum.c.

References `ciL`, `mpi_cmp_abs()`, `mpi_mul_hlp()`, `mpi_sub_hlp()`, `mpi::n`, and `mpi::p`.

Referenced by `mpi_exp_mod()`, and `mpi_montred()`.

### 13.118.2.25 `static void mpi_montred (mpi * A, mpi * N, t_int mm, mpi * T) [static]`

Definition at line 1314 of file `bignum.c`.

References `mpi_montmul()`, `mpi::n`, `mpi::p`, and `mpi::s`.

Referenced by `mpi_exp_mod()`.

### 13.118.2.26 `int mpi_msb (mpi * X)`

Return the number of most significant bits.

Definition at line 200 of file `bignum.c`.

References `biL`, `mpi::n`, and `mpi::p`.

Referenced by `d2i_RSA_PUBKEY()`, `main()`, `mpi_div_mpi()`, `mpi_exp_mod()`, `mpi_gen_prime()`, `mpi_is_prime()`, `mpi_shift_l()`, `mpi_size()`, `mpi_write_string()`, `rsa_check_pubkey()`, `rsa_decryption()`, `rsa_encryption()`, and `rsa_gen_key()`.

### 13.118.2.27 `static void mpi_mul_hlp (int i, t_int * s, t_int * d, t_int b) [static]`

Definition at line 892 of file `bignum.c`.

References `MULADDC_CORE`, `MULADDC_INIT`, and `MULADDC_STOP`.

Referenced by `mpi_montmul()`, and `mpi_mul_mpi()`.

### 13.118.2.28 `int mpi_mul_int (mpi * X, mpi * A, t_int b)`

Baseline multiplication:  $X = A * b$ .

#### Returns:

0 if successful, 1 if memory allocation failed

Definition at line 992 of file `bignum.c`.

References `mpi_mul_mpi()`, `mpi::n`, `mpi::p`, and `mpi::s`.

Referenced by `mpi_div_mpi()`, and `mpi_read_string()`.

### 13.118.2.29 `int mpi_mul_mpi (mpi * X, mpi * A, mpi * B)`

Baseline multiplication:  $X = A * B$ .

#### Returns:

0 if successful, 1 if memory allocation failed

Definition at line 956 of file `bignum.c`.

References MPI\_CHK, mpi\_copy(), mpi\_free(), mpi\_grow(), mpi\_init(), mpi\_lset(), mpi\_mul\_hlp(), mpi::n, mpi::p, and mpi::s.

Referenced by main(), mpi\_gcd(), mpi\_is\_prime(), mpi\_mul\_int(), mpi\_self\_test(), rsa\_check\_privkey(), rsa\_gen\_key(), and rsa\_private().

### 13.118.2.30 int mpi\_read\_binary (mpi \* *X*, unsigned char \* *buf*, int *buflen*)

Import *X* from unsigned binary data, big endian.

#### Parameters:

*X* destination [mpi](#)

*buf* input buffer

*buflen* input buffer size

#### Returns:

0 if successful, 1 if memory allocation failed

Definition at line 484 of file bignum.c.

References CHARS\_TO\_LIMBS, ciL, MPI\_CHK, mpi\_grow(), mpi\_lset(), and mpi::p.

Referenced by asn1\_get\_mpi(), d2i\_RSA\_PUBKEY(), dhm\_read\_bignum(), dhm\_read\_public(), rsa\_private(), and rsa\_public().

### 13.118.2.31 int mpi\_read\_file (mpi \* *X*, int *radix*, FILE \* *fin*)

Read *X* from an opened file.

#### Parameters:

*X* destination [mpi](#)

*radix* input numeric base

*fin* input file handle

#### Returns:

0 if successful, or an XYSSL\_ERR\_MPI\_XXX error code

Definition at line 391 of file bignum.c.

References mpi\_get\_digit(), mpi\_read\_string(), and XYSSL\_ERR\_MPI\_FILE\_IO\_ERROR.

Referenced by main().

### 13.118.2.32 int mpi\_read\_mystring (mpi \* *X*, int *radix*, char \* *s*)

Definition at line 418 of file bignum.c.

References mpi\_get\_digit(), and mpi\_read\_string().

Referenced by rsa\_decryption(), and rsa\_encryption().

**13.118.2.33 int mpi\_read\_string (mpi \* X, int radix, char \* s)**

Import from an ASCII string.

**Parameters:**

*X* destination [mpi](#)  
*radix* input numeric base  
*s* null-terminated string buffer

**Returns:**

0 if successful, or an XYSSL\_ERR\_MPI\_XXX error code

Definition at line 243 of file bignum.c.

References BITS\_TO\_LIMBS, ciL, int, mpi\_add\_int(), MPI\_CHK, mpi\_free(), mpi\_get\_digit(), mpi\_grow(), mpi\_init(), mpi\_lset(), mpi\_mul\_int(), mpi::p, mpi::s, and XYSSL\_ERR\_MPI\_BAD\_INPUT\_DATA.

Referenced by main(), mpi\_read\_file(), mpi\_read\_mystring(), mpi\_self\_test(), rsa\_self\_test(), and ssl\_set\_dh\_param().

**13.118.2.34 int mpi\_self\_test (int verbose)**

Checkup routine.

**Returns:**

0 if successful, or 1 if the test failed

Definition at line 1854 of file bignum.c.

References MPI\_CHK, mpi\_cmp\_mpi(), mpi\_div\_mpi(), mpi\_exp\_mod(), mpi\_free(), mpi\_init(), mpi\_inv\_mod(), mpi\_mul\_mpi(), and mpi\_read\_string().

Referenced by main().

**13.118.2.35 int mpi\_shift\_l (mpi \* X, int count)**

Left-shift:  $X \ll= \text{count}$ .

**Returns:**

0 if successful, 1 if memory allocation failed

Definition at line 526 of file bignum.c.

References biL, BITS\_TO\_LIMBS, MPI\_CHK, mpi\_grow(), mpi\_msb(), mpi::n, and mpi::p.

Referenced by mpi\_div\_mpi(), mpi\_exp\_mod(), and mpi\_gen\_prime().

**13.118.2.36 int mpi\_shift\_r (mpi \* X, int count)**

Right-shift:  $X \gg= \text{count}$ .

**Returns:**

0 if successful, 1 if memory allocation failed

Definition at line 575 of file bignum.c.

References `biL`, `mpi::n`, and `mpi::p`.

Referenced by `dhm_make_params()`, `dhm_make_public()`, `mpi_div_mpi()`, `mpi_gcd()`, `mpi_gen_prime()`, `mpi_inv_mod()`, and `mpi_is_prime()`.

**13.118.2.37 int mpi\_size (mpi \* X)**

Return the total size in bytes.

Definition at line 218 of file bignum.c.

References `mpi_msb()`.

Referenced by `dhm_calc_secret()`, `dhm_make_params()`, `dhm_read_params()`, `mpi_write_binary()`, `x509parse_crt()`, and `x509parse_key()`.

**13.118.2.38 int mpi\_sub\_abs (mpi \* X, mpi \* A, mpi \* B)**

Unsigned subtraction:  $X = |A| - |B|$ .

**Returns:**

0 if successful, `XYSSL_ERR_MPI_NEGATIVE_VALUE` if B is greater than A

Definition at line 761 of file bignum.c.

References `MPI_CHK`, `mpi_cmp_abs()`, `mpi_copy()`, `mpi_free()`, `mpi_init()`, `mpi_sub_hlp()`, `mpi::n`, `mpi::p`, and `XYSSL_ERR_MPI_NEGATIVE_VALUE`.

Referenced by `mpi_add_mpi()`, `mpi_gcd()`, and `mpi_sub_mpi()`.

**13.118.2.39 static void mpi\_sub\_hlp (int n, t\_int \* s, t\_int \* d) [static]**

Definition at line 740 of file bignum.c.

Referenced by `mpi_montmul()`, and `mpi_sub_abs()`.

**13.118.2.40 int mpi\_sub\_int (mpi \* X, mpi \* A, int b)**

Signed subtraction:  $X = A - b$ .

**Returns:**

0 if successful, 1 if memory allocation failed

Definition at line 876 of file bignum.c.

References `mpi_sub_mpi()`, `mpi::n`, `mpi::p`, and `mpi::s`.

Referenced by `main()`, `mpi_gen_prime()`, `mpi_is_prime()`, `rsa_check_privkey()`, and `rsa_gen_key()`.

**13.118.2.41 int mpi\_sub\_mpi (mpi \* *X*, mpi \* *A*, mpi \* *B*)**

Signed subtraction:  $X = A - B$ .

**Returns:**

0 if successful, 1 if memory allocation failed

Definition at line 829 of file bignum.c.

References `mpi_add_abs()`, `MPI_CHK`, `mpi_cmp_abs()`, `mpi_sub_abs()`, and `mpi::s`.

Referenced by `mpi_div_mpi()`, `mpi_inv_mod()`, `mpi_mod_mpi()`, `mpi_sub_int()`, and `rsa_private()`.

**13.118.2.42 void mpi\_swap (mpi \* *X*, mpi \* *Y*)**

Swap the contents of *X* and *Y*.

Definition at line 155 of file bignum.c.

Referenced by `rsa_gen_key()`.

**13.118.2.43 int mpi\_write\_binary (mpi \* *X*, unsigned char \* *buf*, int *buflen*)**

Export *X* into unsigned binary data, big endian.

**Parameters:**

*X* source [mpi](#)

*buf* output buffer

*buflen* output buffer size

**Returns:**

0 if successful, `XYSSL_ERR_MPI_BUFFER_TOO_SMALL` if *buf* isn't large enough

**Note:**

Call this function with `*buflen = 0` to obtain the minimum required buffer size in `*buflen`.

Definition at line 506 of file bignum.c.

References `ciL`, `mpi_size()`, `mpi::p`, and `XYSSL_ERR_MPI_BUFFER_TOO_SMALL`.

Referenced by `dhm_calc_secret()`, `dhm_make_public()`, `rsa_private()`, and `rsa_public()`.

**13.118.2.44 int mpi\_write\_file (char \* *p*, mpi \* *X*, int *radix*, FILE \* *fout*)**

Write *X* into an opened file, or stdout.

**Parameters:**

*p* prefix, can be NULL

*X* source [mpi](#)

*radix* output numeric base



*fout* output file handle

**Returns:**

0 if successful, or an XYSSL\_ERR\_MPI\_XXX error code

**Note:**

Set fout == NULL to print X on the console.

Definition at line 447 of file bignum.c.

References MPI\_CHK, mpi\_write\_string(), and XYSSL\_ERR\_MPI\_FILE\_IO\_ERROR.

Referenced by generate\_RSA\_keys\_ciphertext(), generate\_RSA\_keys\_plaintext(), and main().

**13.118.2.45 static int mpi\_write\_hlp (mpi \* X, int radix, char \*\* p) [static]**

Definition at line 301 of file bignum.c.

References MPI\_CHK, mpi\_cmp\_int(), mpi\_div\_int(), mpi\_mod\_int(), and XYSSL\_ERR\_MPI\_BAD\_INPUT\_DATA.

Referenced by mpi\_write\_string().

**13.118.2.46 int mpi\_write\_string (mpi \* X, int radix, char \* s, int \* slen)**

Export into an ASCII string.

**Parameters:**

*X* source [mpi](#)

*radix* output numeric base

*s* string buffer

*slen* string buffer size

**Returns:**

0 if successful, or an XYSSL\_ERR\_MPI\_XXX error code

**Note:**

Call this function with \*slen = 0 to obtain the minimum required buffer size in \*slen.

Definition at line 328 of file bignum.c.

References ciL, MPI\_CHK, mpi\_copy(), mpi\_free(), mpi\_init(), mpi\_msb(), mpi\_write\_hlp(), mpi::n, mpi::p, mpi::s, XYSSL\_ERR\_MPI\_BAD\_INPUT\_DATA, and XYSSL\_ERR\_MPI\_BUFFER\_TOO\_SMALL.

Referenced by mpi\_write\_file().

## 13.118.3 Variable Documentation

**13.118.3.1 const int small\_prime[] [static]**

**Initial value:**

```
{
    3,    5,    7,   11,   13,   17,   19,   23,
    29,   31,   37,   41,   43,   47,   53,   59,
    61,   67,   71,   73,   79,   83,   89,   97,
    101,  103,  107,  109,  113,  127,  131,  137,
    139,  149,  151,  157,  163,  167,  173,  179,
    181,  191,  193,  197,  199,  211,  223,  227,
    229,  233,  239,  241,  251,  257,  263,  269,
    271,  277,  281,  283,  293,  307,  311,  313,
    317,  331,  337,  347,  349,  353,  359,  367,
    373,  379,  383,  389,  397,  401,  409,  419,
    421,  431,  433,  439,  443,  449,  457,  461,
    463,  467,  479,  487,  491,  499,  503,  509,
    521,  523,  541,  547,  557,  563,  569,  571,
    577,  587,  593,  599,  601,  607,  613,  617,
    619,  631,  641,  643,  647,  653,  659,  661,
    673,  677,  683,  691,  701,  709,  719,  727,
    733,  739,  743,  751,  757,  761,  769,  773,
    787,  797,  809,  811,  821,  823,  827,  829,
    839,  853,  857,  859,  863,  877,  881,  883,
    887,  907,  911,  919,  929,  937,  941,  947,
    953,  967,  971,  977,  983,  991,  997, -103
}
```

Definition at line 1639 of file bignum.c.

Referenced by mpi\_is\_prime().

## 13.119 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/library/certs.c File Reference

```
#include "xyssl/config.h"
```

### Variables

- char `test_ca_crt` []
- char `test_ca_key` []
- char `test_ca_pwd` [] = "test"
- char `test_srv_crt` []
- char `test_srv_key` []
- char `test_cli_crt` []
- char `test_cli_key` []
- char `xyssl_ca_crt` []

### 13.119.1 Variable Documentation

#### 13.119.1.1 char test\_ca\_crt[]

Initial value:

```
"-----BEGIN CERTIFICATE-----\r\n"
"MIIDpTCCAo2gAwIBAgIBADANBgkqhkiG9w0BAQUFADBFMQswCQYDVQQGEwJGUjEO\r\n"
"MAwGA1UEBxMFUGFyaXMxDjAMBGNVBAoTBVh5U1NMMRYwFAYDVQQDEw1YeVNTTCBU\r\n"
"ZXN0IENBMB4XDTA3MDcwNzA1MDAxOFoXDTE3MDcwNzA1MDAxOFowRTElMAkGA1UE\r\n"
"BhMCRlIxZjAMBGNVBACTBVBhcm1zMQ4wDAYDVQQKEwVYeVNTTDEWMBQGA1UEAxMN\r\n"
"WH1TU0wgVGZvdCBBDQTCASiWdQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAM+k\r\n"
"gt70fIPiYqmvXr+9uPmWoN405eoSzxdiRLLCqL4V/Ts0E/H+JNfHS4Dh1AgxrJu\r\n"
"+ZiadvSJuHkI6eliMkAh5SU1DqaF3jrrFdJCooM6a077M4CRkE1tdAeZdf+BYp0q\r\n"
"BeMU9Y2+j7ibsQaPAizunbXLf4QdteExCwhYRJ8OVXSEasNt339gJzTD6kOhES3b\r\n"
"lEN3qb61qFaJ5MLHTix5uNVc2rvbOizV5oLhJNqm52AKOp1ltv6WTiI8loagvAc\r\n"
"jlhEZRNB9e15SL6Jai/uFcqXKzfXNKW3FYpQHfGobmiMfGt1lUSBj3F2mrqEC7gC\r\n"
"whY/FDvAI64/k5LZAFkCAwEAaOBnzCBnDAMBGNVHRMEBTADAQH/MB0GA1UdDgQW\r\n"
"BBS87h+Y6Porg+SkfV7DdXKTMdkyZzBtBgNVHSMZjBkgBS87h+Y6Porg+SkfV7D\r\n"
"dXKTMdkyZ6FJpEcwRTElMAkGA1UEBhMCRlIxZjAMBGNVBAoTBVBhcm1zMQ4wDAYD\r\n"
"VQKKEwVYeVNTTDEWMBQGA1UEAxMNWH1TU0wgVGZvdCBBDQYIBADANBgkqhkiG9w0B\r\n"
"AQUFAAOCAQEAIHdohONCg6KAAhWDSmfEgSbKUI8/Zr/d56uw42H00sj/uKPQzUco\r\n"
"3Mx2BYElm1itg7q5OhrkB7J4ZB78EtNZM84nV+y6od3YpR0Z9VUxCx7948MozYRy\r\n"
"TKF5x/lKHXlPJkFE04clKdWTFAtWtGhewXrHJQ8C+ENh2Up2wTVh3Z+pEzuZNv3\r\n"
"u/JYu1H+vkt31lWCy/9mxUnu+anW1DzxPwnjy41x6Mi0BD2qfKBWLjVS+7v6ALcj\r\n"
"S2oRWwr4LUvXT7z9BBAvw2eJQD+a4uAya6EURG7AsAvr5MnWn/r0wLWmBJ6fBlYp\r\n"
"FlkOmamOFvstLMf74rLX+LGKeJ/nwui5FQ==\r\n"
"-----END CERTIFICATE-----\r\n"
```

Definition at line 25 of file certs.c.

Referenced by `main()`, `ssl_test()`, and `x509_self_test()`.

#### 13.119.1.2 char test\_ca\_key[]

Definition at line 49 of file certs.c.

Referenced by `x509_self_test()`.

### 13.119.1.3 char test\_ca\_pwd[ ] = "test"

Definition at line 81 of file certs.c.

Referenced by x509\_self\_test().

### 13.119.1.4 char test\_cli\_crt[ ]

**Initial value:**

```
"-----BEGIN CERTIFICATE-----\r\n"
"MIIDPTCAiWgAwIBAgIBATANBgkqhkiG9w0BAQUFADBFMQswCQYDVQQGEwJGUjE0\r\n"
"MAwGA1UEBxMFUGFyaXMxdjAMBgNVBAoTBVh5U1NMMRYwFAYDVQQDEw1YeVNTTCBU\r\n"
"ZXN0IENBMB4XDTA3MDcwNzA1MDEyMFoXDTA4MDcwNjA1MDEyMFowMDELMakGA1UE\r\n"
"BhMCR1IxdjAMBgNVBAoTBVh5U1NMMREwDwYDVQQDEwhKb2UgVXNlcjCCASIdQYJ\r\n"
"KoZiIhvcNAQEBBQADgGEPADCCAQoCggEBAKZkg8ANl6kGmLGqKc6KUHb9IsZwb2+K\r\n"
"jBw83Qb0KuvPVnu3MzEcXfVOZ83g0PL/z8ob5PKr8HP6bVYzhsD65imcCDEEVPk2\r\n"
"9x0XGTggGjB601Fd8aTShUWE4NLrKw6YNXTXgTndvhHNxXwqmdNVLkmZjj3ZwYUc\r\n"
"eEE8eE5jHs8cMDXJLMCwgKIM7Sax220hSHQHKwifVO4/Fdw5G+Suys8PhMX2jDXM\r\n"
"ICFwq8ld+bZGoNUtgp48FWhAMfJyTEaHh9LC46KkqGSDRIzx7/4cPB6QqrpzJN0o\r\n"
"Kr8kh7vdRDTFDm023D4C5l0Bw/2aC76DhEJpB2bGA4iIszJs+F/PIL8CAwEAAaNN\r\n"
"MEswCQYDVROTBAlwADAdBgNVHQ4EFgQUiWX1IvjrDYgt0zz5Sq16x01k0o4wHwYD\r\n"
"VR0jBBBgwFoAUvO4fmOj6K4PkpH1ew3VykzHZMmcwDQYJKoZIhvcNAQEFBQADggEB\r\n"
"AGdqD7VthJmC+oeemuHk2TQX2wZNU+GsC+RLjtlencnky95KnljGvMtCznyLkS5D\r\n"
"faJLKfR1No8pk5GrdsccgyIuQx5WnHNv4QBZmMsmvDICxzRQaxuPFHbS4aLXldL\r\n"
"yOWm524qkMHPCKvA86blYsEkksGDV47fF9ZkOQ8nkh7Z4eY4/5TwqTY72ww5g4NL\r\n"
"6DZtWpGpGbx99NRANvzcq9D+ElxkgHnH4YWafOKBclSgqrutbRLi2uZx/QpvuF+i\r\n"
"sUbe+HFPmWuU5lBv/oOhQkz0VD+HusYtXWS2lG88ct40aNLy2CkYUugdTR/b9Uea\r\n"
"p/i862sL/1040qlQ0xv5N7U=\r\n"
"-----END CERTIFICATE-----\r\n"
```

Definition at line 134 of file certs.c.

Referenced by main(), and x509\_self\_test().

### 13.119.1.5 char test\_cli\_key[ ]

**Initial value:**

```
"-----BEGIN RSA PRIVATE KEY-----\r\n"
"MIIEowIBAAKCAQEApmsDwa2XqQaYsaopzopQdv0ixnBvb4qMHDzdBvQq689We7cz\r\n"
"MRxcW85nzeDQ8v/Pyhvk8qvw/ptVjOGwPrmKZwIMQRU+Tb2vRcZOCaAMHrTUV3x\r\n"
"pNKFRTg0usrDpgldNeBMl2+Ec3FfCqZ0lUuSZmOPdnBhRwxQTx4TmMezxwNcks\r\n"
"wLCAogztJrHbY6FIdAcrCJ9U7j8V3Dkb5K7Kzw+ExfaMNCwgIXCryV35tkag1S2C\r\n"
"njwVaEAX8nJMRoeH0sLjoqSoZINEjPHv/hw8HpCqunMk3SgqvYQfu91ENMUOY7bc\r\n"
"PgLMXQHD/ZoLvoOEQmKHZsYDiIizMmz4X88gvwIDAQABAoIBAEOBnBKAjDVN+j4ax\r\n"
"lDjEwZKqxVkmAUXDBDyDrCjxRoWY2gz7YW1ALUMUbeV0f05v1zVrwbkUKKZeVBx\r\n"
"QA9zRw28H8A6tfvolHgrIcx4dixMh3ePC+DVDJ6zglvKV2ipAwBufKYIrX0r4Io2\r\n"
"ZqUrNg9CeEYNlkHWceaN12rhYwO82pgHxnB1p5pI42pY7lzyLgSddf5n+M5UBOJI\r\n"
"gsNCKvbGdv7WQPVFTRDiRgEnCJ3rI8oPSK6MOUWJw3rh2hbKx+ex8NPvEKbzEXiU\r\n"
"p5j1AlbHIWP5sYBbAlYviFtryAV4fyfLcWpfoqa33Oozofjlwoj0Aixz+6rerLjZ\r\n"
"cpTSrAECgYEA2oCffUo6HH3Lq9oeWhFCOyG3YjZmFrJaJwJHnvroX9/pXHqYKog\r\n"
"TeHcjUJBtFzW0klcetYbZCFqT8v9nf0uPlgaiVGctXf1MSbFXDUFKkYBiFwzdWMT\r\n"
"Ysmvfhf82jMWZecsXTyDRL858R5WPZ52qEsCc5X2un7QENm6FtVT8CgYEAwwKS\r\n"
"zQNzuoJETqZX7AalMk3JM8Fdam+Qm5LNMcbvbkKI8HKMS1VMuqaR0XdAX/imXAx\r\n"
"PlVhSsmoSDbsMpxBEZlptpCen/GcqqITxANTakrBHxqb2aQ5EEu7SgzfHZWse3\r\n"
"vQEYfctFB1PdcdZUDzk4/w7WmyivpYtCWoaHlIECgYEA0UYZ+1UJfVpapRj+swMP\r\n"
"DrQbo7i7t7lUaFYLNpFX2OPLTWC5txqn1OruTu5VHDqE+5hneDNUUTT3uOg4B2q\r\n"
"mdmmaNjh2M6wz0e0BVfexhNQynqMaqTe32IOM8DFs3L0xacgg7JfVn6P7CeQGOVe\r\n"
"wc96kICw6ZxhtJSqpOGipt8CgYBI/OPw+IXxJK4nNSpe+u4vCYP5mUI9hKEFYCbt\r\n"
"qKwvyAUknn/zgiIQ+z/iSErFMPmlwXjvWi0gL/qPb+Pp4hCLX8u2zNhY08P4Gin\r\n"
```

### 13.119 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/library/certs.c File Reference 809

```
"Ej+pAntWxq+kHyfKEI5dyRwV/snfvlqwjy404JsSF3VMhIMdYDPzbb72Qnni5w5l\r\n"
"j00eAQKBgBqt9jJMd1JdpemC2dm0BuuDIz2h3/MH+CMjfaDLenVpKykn17B6N92h\r\n"
"k1MesqK3RQZDGwauDw431LQw0R69onn9fCM3wJw2yEC6wC9sF8I8hsNZbt64yZhZ\r\n"
"4Bi2YRTiHhpEuBqKlHHLDFHneo3SMYh8PU/PDQQcyWGHHUi9z1RE\r\n"
"-----END RSA PRIVATE KEY-----\r\n"
```

Definition at line 156 of file certs.c.

Referenced by main().

#### 13.119.1.6 char test\_srv\_cert[]

Initial value:

```
"-----BEGIN CERTIFICATE-----\r\n"
"MIIDPjCCAiaGAWIBAgIBA jANBgkqhkiG9w0BAQUFADBFMQswCQYDVQQGEwJGU jEO\r\n"
"MAwGA1UEBxMFUGFyaXMxD jAMBgNVBAoTBVh5U1NMMRYwFAyDVQQDEw1YeVNTTCBU\r\n"
"ZXN0IENBMB4XDTA3MDcwNzA1MDEyOVoxDTA4MDcwNjA1MDEyOVowMTELMakGA1UE\r\n"
"BhMCRlI xD jAMBgNVBAoTBVh5U1NMMRIwEAYDVQQDEw1sb2NhbGhvc3QwgGEiMA0G\r\n"
"CSqGSIB3DQEBAQUAA4IBDwAwggEKAoIBAQC40PDcGTgmHkt6noXDfk jVuymjiNYB\r\n"
"gttiL7uAlKe3tXStacEecQek/OJxYqYr7ffcWalS29LL6HbKpi0xLZKBbD9ACkDh\r\n"
"1Z/SvHlyQPILJdYb9DMw+kzZds5myXUjzn7Aem1YjoxMZUAMyc34i2900X2pL0v2\r\n"
"SfCeJ9Ym4MOnZxYl217+dX9ZbkGIgrT6uY2IYK4boDwxBTcyT8i/NPsVs iMwtWPM\r\n"
"rnQMr+XbgS98sUzcZE70Pe1TlV9Iy8j/8d2OiFo+qTyMu/6UpM2s3gdkQkMzx+Sm\r\n"
"4QitRUjzmEXeUePRUjEgHiv7vz069xuvBzrks36w5BXiVAhLke/OTKVPagMBAAGj\r\n"
"TTBLMAkGA1UdEwQCMAAwHQYDVROBBYEFNkOyCTx64SDdPySGW1/tzD7/WMSMB8G\r\n"
"AlUdIwQYMBaAF LzuH5 jo+iuD5KR9XsN1cpMx2TJnMA0GCSqGSIB3DQEBBQUAA4IB\r\n"
"AQBelJv5t+suagy5Lo5b jNeHjNZfgg8EigDQ7Nqaosv1QZAsh2N34Gg5YdkGyVdg\r\n"
"s32I/K5aaywyUbG9qVXQxCM2T95qBqyK56h9yJoZKWQD9H//+zB8kCK/16WvRfv3\r\n"
"V7eSR19qOFwLHe+1qGh2YhxeDUfYi+fm4D36dGxqC2A34tzjo0QPHKtIeqM0kJy\r\n"
"zzL65T1bJQKkyTurHofFv0jW9ZFG2wkGysVgCY5fjuLi1do/sWUaXd2987iNFa+K\r\n"
"FrHsTi6urSfZuG1ZNxDXDHEE7Q2snAvvev+KR7DD9X4DJGcPX9gA4CGJj+9ZzyAA\r\n"
"ZTGpOzk1hIH44RFs2lJMZR1E\r\n"
"-----END CERTIFICATE-----\r\n"
```

Definition at line 83 of file certs.c.

Referenced by main(), and ssl\_test().

#### 13.119.1.7 char test\_srv\_key[]

Initial value:

```
"-----BEGIN RSA PRIVATE KEY-----\r\n"
"MIIEowIBAAKCAQEaAuNDw3Bk4Jh5Lep6Fw35I1bsp04jWAYI7Yi+7gNSnt7V0rWnB\r\n"
"HnEHpPzicWKmK+333FmpUtvSy+h2yqYtMS2SgWw/QApA4dWf0rx5ckDyCyXWG/Qz\r\n"
"MPpM2XbOZs11I85+wHptWI6MTGVADMnN+ItvdNF9qS9L9knwni fWJuDDp2cWJdte\r\n"
"/nV/WW5ICIK0+rmNiGCuG6A8MW03Mk/IvzT7FbI jMLVjzK50DK/124EvfLFM3GRO\r\n"
"9D3tU5VfSMvI/ /HdjohaPqk8 jLv+1KTNrN4HZEJDM8fkpuEIrUVI85hF31Hj0VIx\r\n"
"IBYl+7890vcblQc65LN+sOQV4lQIS5HvzkylTwIDAQABAoIBABeah8h0aBlmMRmd\r\n"
"+VN4Y3D4kF7UcRCMQ21Mz1Oq1Si/QgGLyiBLK0DFE16LzNE7eTZpNRjh/1AQhmtn\r\n"
"QcpQGa/xlTom1RbCo8DUVWZkKQWHdYroa01MDliPt dimzhEepE2M1T5EJmLzY3S+\r\n"
"qVGe7UMsJjJfWgJAezyXteANQK+2YSt+CjPIqIHch1KexUnvdN9++1oEx6AbuZ8T\r\n"
"4avhFYZQP15tZNGsk2LfQlYS/NfbowkCsd0/TVubJBmDGUML/E5MbxjxLz1aNB2M\r\n"
"V59cBNngsgA35CODAUf4xOyoSfZGqG1Rb9qQrv1E6Jz56dG8SsKF3HqnDjxiPOVBN\r\n"
"FBnVJ+ECgYEA29MhAsKmm4XqBUKp6pIMFTgm/s1E5vxig70vqiL+guvBhhQ7zs1\r\n"
"8UMTNXx0MELNoB/ev9fN0Cjc1Vr46b/x/yDw7wMb96i+vzEN0zu4RHWi3OWpCPbp\r\n"
"qBKEi3hzN8M+BulPX8CDQx3aLRrfxw51J5EuA0NeybngbItgxTi0u6kCgYEA1zr0\r\n"
"6P5YdOhYHTSWD1ked49MApcVuzaHnsHZVAhUqu3Rwiy9LRaJLZfr7fQDb9DYJbZp\r\n"
"sxTRLG6LSAcSR7mw+m+GvNqGt/9pSqbtW+L/VwVWSyF+YYklxZUD3UAAyrDVcDEC\r\n"
"a5S+jad4Csi/1Vht5ulWICKwL1fJvadnSubKNDCcGYA+71xVGPP+1sFgTiytfrC8\r\n"
```

```
"5n2rl4MxinJ9+w0I+EbZCKNMYGvTgiU4dJasSMEdiBKslFMGo7dF8F0BLHF1IsIa\r\n"
"5Aht2XItXn9l54o90iTQXMmK6qmRaneM6fhOoeaCwYAhpGxYIpx/Xr4TOhiag46\r\n"
"jMMaphAeOvw4t1K2RDziOQKBgQCYPCCU0gxuw/o1jda2CxbZy9EmU/erEX09+0n+\r\n"
"TOFQpSEPq/z9WaxAFY9LfsdZ0ZktoeHmalbNdL3i6A3DWAM3YSQzQMRPmzOWnqXx\r\n"
"cgoCBmlvzkzaeLjO5phMoLQHJmmafvuCG6uxov3F8Hi3LyHUF2c8k0nL6ucmJ3vj\r\n"
"uzu4AQKBgBSASMAJS63M9UJB1Eazy2v2Nww04CmzNxUfWrHuKpd/C2ik4QKu0sRO\r\n"
"r9KnkDgxxEhjDm7lXhlW12PU42yORst5I3EaalCfi4KPFn/ozt+iNBYrzd8Tyvnb\r\n"
"qkdEC10+G2Fo/ER4NRCv7a24WNEsOMGzGRqw5cnSJrjzbZLYMaIyK\r\n"
"-----END RSA PRIVATE KEY-----\r\n"
```

Definition at line 105 of file certs.c.

Referenced by main(), and ssl\_test().

### 13.119.1.8 char xyssl\_ca\_cert[]

**Initial value:**

```
"-----BEGIN CERTIFICATE-----\r\n"
"MIID4DCCAsigAwIBAgIJAOLw9BMV1jxMMA0GCSqGSIb3DQEBBQUAMFMCzAJBgNV\r\n"
"BAYTAKZSMQ4wDAYDVQQIEwVQYXJpczEOMAwGA1UEChMFWH1TU0wxJDAiBgNVBAMT\r\n"
"G1h5U1NMIENlcnRpZmljYXRlIEF1dGhvcml0eTAeFw0wNjEwMzEyMjU5MjRaFw0x\r\n"
"NjEwMzEyMjU5MjRaMFMCzAJBgNVBAYTAKZSMQ4wDAYDVQQIEwVQYXJpczEOMAwG\r\n"
"A1UEChMFWH1TU0wxJDAiBgNVBAMTG1h5U1NMIENlcnRpZmljYXRlIEF1dGhvcml0\r\n"
"eTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAKnprf1RQ7IYPI3FmI/h\r\n"
"f2EJGfaiP+Jt551VZWfRs3A56Nn4KS57zTAKZUA7YGBLwLAznfaphJ7SvENALeZR\r\n"
"/J0c/n9jwMpfXReQL7RVpgg/zlR+t2DUI3DAwigPZiHHCSJSBC73vpMc6uH0eV2d\r\n"
"itqjJUnJG5F8Zg9/gX4UMRADlwGWqxvs+jc9i0XFKOEzga8+rOnE6WvKyBM4e20I\r\n"
"HcO4BPF92d6sm4qLgyR4oXUkBz6NfDWX8ZdTvXuRaK9qMy1327cCT48sis9F6/eK\r\n"
"QAx0VHlqGTtxDYjHJPsmLNejuyUDvsaC8TMCSpFTXpMTUvJdmxsF1LZWmIo5lY5\r\n"
"zh0CAwEAAaOBtjCBszAdBgNVHQ4EFgQUBlVzoNjrkkgPkj8xjuFvVP2E4GswgYMG\r\n"
"AlUdIwR8MHqAFaZVc6DSa5JID5CFMY7hb1T9hOBroVekVTBTMQswCQYDVQQGEwJG\r\n"
"UjEOMAwGA1UECBMFUGFyaXMxDjAMBGNVBAoTBVh5U1NMMSQwIgYDVQQDEXTYeVNT\r\n"
"TCBDZXJ0aWZpY2F0ZSBBDXR0b3JpdHmCCQDi8PQTFdY8TDAMBGNVHRMEBTADAQH\r\n"
"MA0GCSqGSIb3DQEBBQUAA4IBAQCgD65b2l5BASFSpVlrcRnLZu/99eWTVAJwJbbD\r\n"
"VhPAQiET0W4U/85EDK7uoFo/SEjyMB/m4T20A8FIDaK7jBPo/1gtbuQjGMR17h+z\r\n"
"F2iGuNh26Td26Uzqclt3oiFtSvDRoZ/9kqkEy7Lrs7FBzOmvfTvrqvADf7cLMA2D\r\n"
"ri/otDpPr4XoDnwd4C+4bQC/Gr3Uder4VAeTOJtKdGqfYlvPwPSPVBDuVLUybKi\r\n"
"8cMAT6p9IG1e12u6vFqcBT/I67Q0bGU6gzVVz9ZVULXOYzMjjLafVXC1gesUH2WT\r\n"
"gTEAnEBkSRrkfAi+RezoEFAbmEl3fPt09dwSPku3x7cB3zaJ\r\n"
"-----END CERTIFICATE-----\r\n"
```

Definition at line 185 of file certs.c.

Referenced by main().

## 13.120 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/library/debug.c File Reference

```
#include "xyssl/config.h"
#include "xyssl/debug.h"
#include <stdarg.h>
#include <stdlib.h>
```

### Functions

- char \* [debug\\_fmt](#) (const char \*format,...)
- void [debug\\_print\\_msg](#) (ssl\_context \*ssl, int level, char \*file, int line, char \*text)
- void [debug\\_print\\_ret](#) (ssl\_context \*ssl, int level, char \*file, int line, char \*text, int ret)
- void [debug\\_print\\_buf](#) (ssl\_context \*ssl, int level, char \*file, int line, char \*text, unsigned char \*buf, int len)
- void [debug\\_print\\_mpi](#) (ssl\_context \*ssl, int level, char \*file, int line, char \*text, mpi \*X)
- void [debug\\_print\\_cert](#) (ssl\_context \*ssl, int level, char \*file, int line, char \*text, x509\_cert \*cert)

### 13.120.1 Function Documentation

#### 13.120.1.1 char\* debug\_fmt (const char \*format, ...)

Definition at line 38 of file debug.c.

#### 13.120.1.2 void debug\_print\_buf (ssl\_context \*ssl, int level, char \*file, int line, char \*text, unsigned char \*buf, int len)

Definition at line 82 of file debug.c.

References `_ssl_context::f_dbg`, and `_ssl_context::p_dbg`.

#### 13.120.1.3 void debug\_print\_cert (ssl\_context \*ssl, int level, char \*file, int line, char \*text, x509\_cert \*cert)

Definition at line 169 of file debug.c.

References `debug_print_mpi()`, `rsa_context::E`, `_ssl_context::f_dbg`, `rsa_context::N`, `_x509_cert::next`, `_ssl_context::p_dbg`, `_x509_cert::rsa`, and `x509parse_cert_info()`.

#### 13.120.1.4 void debug\_print\_mpi (ssl\_context \*ssl, int level, char \*file, int line, char \*text, mpi \*X)

Definition at line 124 of file debug.c.

References `_ssl_context::f_dbg`, `mpi::n`, `mpi::p`, and `_ssl_context::p_dbg`.

Referenced by `debug_print_cert()`.

**13.120.1.5 void debug\_print\_msg (ssl\_context \* *ssl*, int *level*, char \* *file*, int *line*, char \* *text*)**

Definition at line 52 of file debug.c.

References `_ssl_context::f_dbg`, and `_ssl_context::p_dbg`.

**13.120.1.6 void debug\_print\_ret (ssl\_context \* *ssl*, int *level*, char \* *file*, int *line*, char \* *text*, int *ret*)**

Definition at line 66 of file debug.c.

References `_ssl_context::f_dbg`, and `_ssl_context::p_dbg`.



## 13.121 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/library/des.c File Reference

```
#include "xyssl/config.h"
#include "xyssl/des.h"
#include <string.h>
#include <stdio.h>
```

### Defines

- #define [GET\\_ULONG\\_BE](#)(n, b, i)
- #define [PUT\\_ULONG\\_BE](#)(n, b, i)
- #define [DES\\_IP](#)(X, Y)
- #define [DES\\_FP](#)(X, Y)
- #define [DES\\_ROUND](#)(X, Y)
- #define [SWAP](#)(a, b) { unsigned long t = a; a = b; b = t; t = 0; }

### Functions

- static void [des\\_setkey](#) (unsigned long SK[32], unsigned char key[8])
- void [des\\_setkey\\_enc](#) ([des\\_context](#) \*ctx, unsigned char key[8])  
*DES key schedule (56-bit, encryption).*
- void [des\\_setkey\\_dec](#) ([des\\_context](#) \*ctx, unsigned char key[8])  
*DES key schedule (56-bit, decryption).*
- static void [des3\\_set2key](#) (unsigned long esk[96], unsigned long dsk[96], unsigned char key[16])
- void [des3\\_set2key\\_enc](#) ([des3\\_context](#) \*ctx, unsigned char key[16])  
*Triple-DES key schedule (112-bit, encryption).*
- void [des3\\_set2key\\_dec](#) ([des3\\_context](#) \*ctx, unsigned char key[16])  
*Triple-DES key schedule (112-bit, decryption).*
- static void [des3\\_set3key](#) (unsigned long esk[96], unsigned long dsk[96], unsigned char key[24])
- void [des3\\_set3key\\_enc](#) ([des3\\_context](#) \*ctx, unsigned char key[24])  
*Triple-DES key schedule (168-bit, encryption).*
- void [des3\\_set3key\\_dec](#) ([des3\\_context](#) \*ctx, unsigned char key[24])  
*Triple-DES key schedule (168-bit, decryption).*
- void [des\\_crypt\\_ecb](#) ([des\\_context](#) \*ctx, unsigned char input[8], unsigned char output[8])  
*DES-ECB block encryption/decryption.*
- void [des\\_crypt\\_cbc](#) ([des\\_context](#) \*ctx, int mode, int length, unsigned char iv[8], unsigned char \*input, unsigned char \*output)  
*DES-CBC buffer encryption/decryption.*

- void [des3\\_crypt\\_ecb](#) ([des3\\_context](#) \*ctx, unsigned char input[8], unsigned char output[8])  
*3DES-ECB block encryption/decryption*
- void [des3\\_crypt\\_cbc](#) ([des3\\_context](#) \*ctx, int mode, int length, unsigned char iv[8], unsigned char \*input, unsigned char \*output)  
*3DES-CBC buffer encryption/decryption*
- int [des\\_self\\_test](#) (int verbose)

## Variables

- static const unsigned long [SB1](#) [64]
- static const unsigned long [SB2](#) [64]
- static const unsigned long [SB3](#) [64]
- static const unsigned long [SB4](#) [64]
- static const unsigned long [SB5](#) [64]
- static const unsigned long [SB6](#) [64]
- static const unsigned long [SB7](#) [64]
- static const unsigned long [SB8](#) [64]
- static const unsigned long [LHs](#) [16]
- static const unsigned long [RHs](#) [16]
- static const unsigned char [des3\\_test\\_keys](#) [24]
- static const unsigned char [des3\\_test\\_iv](#) [8]
- static const unsigned char [des3\\_test\\_buf](#) [8]
- static const unsigned char [des3\\_test\\_ecb\\_dec](#) [3][8]
- static const unsigned char [des3\\_test\\_ecb\\_enc](#) [3][8]
- static const unsigned char [des3\\_test\\_cbc\\_dec](#) [3][8]
- static const unsigned char [des3\\_test\\_cbc\\_enc](#) [3][8]

## 13.121.1 Define Documentation

### 13.121.1.1 #define DES\_FP(X, Y)

Value:

```
{
    X = ((X << 31) | (X >> 1)) & 0xFFFFFFFF; \
    T = (X ^ Y) & 0xAAAAAAAA; X ^= T; Y ^= T; \
    Y = ((Y << 31) | (Y >> 1)) & 0xFFFFFFFF; \
    T = ((Y >> 8) ^ X) & 0x00FF00FF; X ^= T; Y ^= (T << 8); \
    T = ((Y >> 2) ^ X) & 0x33333333; X ^= T; Y ^= (T << 2); \
    T = ((X >> 16) ^ Y) & 0x0000FFFF; Y ^= T; X ^= (T << 16); \
    T = ((X >> 4) ^ Y) & 0x0F0F0F0F; Y ^= T; X ^= (T << 4); \
}
```

Definition at line 257 of file des.c.

Referenced by [des3\\_crypt\\_ecb\(\)](#), and [des\\_crypt\\_ecb\(\)](#).

### 13.121.1.2 #define DES\_IP(X, Y)

**Value:**

```
{
    T = ((X >> 4) ^ Y) & 0x0F0F0F0F; Y ^= T; X ^= (T << 4); \
    T = ((X >> 16) ^ Y) & 0x0000FFFF; Y ^= T; X ^= (T << 16); \
    T = ((Y >> 2) ^ X) & 0x33333333; X ^= T; Y ^= (T << 2); \
    T = ((Y >> 8) ^ X) & 0x00FF00FF; X ^= T; Y ^= (T << 8); \
    Y = ((Y << 1) | (Y >> 31)) & 0xFFFFFFFF; \
    T = (X ^ Y) & 0xAAAAAAAA; Y ^= T; X ^= T; \
    X = ((X << 1) | (X >> 31)) & 0xFFFFFFFF; \
}
```

Definition at line 243 of file des.c.

Referenced by des3\_crypt\_ecb(), and des\_crypt\_ecb().

### 13.121.1.3 #define DES\_ROUND(X, Y)

**Value:**

```
{
    T = *SK++ ^ X; \
    Y ^= SB8[ (T & 0x3F) ^ \
        SB6[ (T >> 8) & 0x3F ] ^ \
        SB4[ (T >> 16) & 0x3F ] ^ \
        SB2[ (T >> 24) & 0x3F ] ]; \
    T = *SK++ ^ ((X << 28) | (X >> 4)); \
    Y ^= SB7[ (T & 0x3F) ^ \
        SB5[ (T >> 8) & 0x3F ] ^ \
        SB3[ (T >> 16) & 0x3F ] ^ \
        SB1[ (T >> 24) & 0x3F ] ]; \
}
```

Definition at line 271 of file des.c.

Referenced by des3\_crypt\_ecb(), and des\_crypt\_ecb().

### 13.121.1.4 #define GET\_ULONG\_BE(n, b, i)

**Value:**

```
{
    (n) = ( (unsigned long) (b)[(i) << 24] \
        | ( (unsigned long) (b)[(i) + 1] << 16) \
        | ( (unsigned long) (b)[(i) + 2] << 8) \
        | ( (unsigned long) (b)[(i) + 3] ) ); \
}
```

Definition at line 39 of file des.c.

Referenced by des3\_crypt\_ecb(), des\_crypt\_ecb(), des\_setkey(), sha1\_process(), and sha2\_process().

### 13.121.1.5 #define PUT\_ULONG\_BE(n, b, i)

**Value:**

```

{
    (b) [(i)      ] = (unsigned char) ( (n) >> 24 ); \
    (b) [(i) + 1] = (unsigned char) ( (n) >> 16 ); \
    (b) [(i) + 2] = (unsigned char) ( (n) >> 8  ); \
    (b) [(i) + 3] = (unsigned char) ( (n)          ); \
}

```

Definition at line 49 of file des.c.

Referenced by des3\_crypt\_ecb(), des\_crypt\_ecb(), sha1\_finish(), and sha2\_finish().

### 13.121.1.6 #define SWAP(a, b) { unsigned long t = a; a = b; b = t; t = 0; }

Definition at line 286 of file des.c.

Referenced by des\_setkey\_dec().

## 13.121.2 Function Documentation

### 13.121.2.1 void des3\_crypt\_cbc (des3\_context \* ctx, int mode, int length, unsigned char iv[8], unsigned char \* input, unsigned char \* output)

3DES-CBC buffer encryption/decryption

#### Parameters:

*ctx* 3DES context  
*mode* DES\_ENCRYPT or DES\_DECRYPT  
*length* length of the input data  
*iv* initialization vector (updated after use)  
*input* buffer holding the input data  
*output* buffer holding the output data

Definition at line 593 of file des.c.

References des3\_crypt\_ecb(), and DES\_ENCRYPT.

Referenced by des\_self\_test(), main(), ssl\_decrypt\_buf(), ssl\_encrypt\_buf(), and x509\_des3\_decrypt().

### 13.121.2.2 void des3\_crypt\_ecb (des3\_context \* ctx, unsigned char input[8], unsigned char output[8])

3DES-ECB block encryption/decryption

#### Parameters:

*ctx* 3DES context  
*input* 64-bit input block  
*output* 64-bit output block

Definition at line 552 of file des.c.

References DES\_FP, DES\_IP, DES\_ROUND, GET\_ULONG\_BE, PUT\_ULONG\_BE, and des3\_context::sk.

Referenced by des3\_crypt\_cbc(), and des\_self\_test().

**13.121.2.3 static void des3\_set2key (unsigned long *esk*[96], unsigned long *dsk*[96], unsigned char *key*[16]) [static]**

Definition at line 381 of file des.c.

References des\_setkey().

Referenced by des3\_set2key\_dec(), and des3\_set2key\_enc().

**13.121.2.4 void des3\_set2key\_dec (des3\_context \* *ctx*, unsigned char *key*[16])**

Triple-DES key schedule (112-bit, decryption).

**Parameters:**

*ctx* 3DES context to be initialized

*key* 16-byte secret key

Definition at line 420 of file des.c.

References des3\_set2key(), and des3\_context::sk.

Referenced by des\_self\_test().

**13.121.2.5 void des3\_set2key\_enc (des3\_context \* *ctx*, unsigned char *key*[16])**

Triple-DES key schedule (112-bit, encryption).

**Parameters:**

*ctx* 3DES context to be initialized

*key* 16-byte secret key

Definition at line 409 of file des.c.

References des3\_set2key(), and des3\_context::sk.

Referenced by des\_self\_test().

**13.121.2.6 static void des3\_set3key (unsigned long *esk*[96], unsigned long *dsk*[96], unsigned char *key*[24]) [static]**

Definition at line 428 of file des.c.

References des\_setkey().

Referenced by des3\_set3key\_dec(), and des3\_set3key\_enc().

**13.121.2.7 void des3\_set3key\_dec (des3\_context \* *ctx*, unsigned char *key*[24])**

Triple-DES key schedule (168-bit, decryption).

**Parameters:**

*ctx* 3DES context to be initialized

*key* 24-byte secret key

Definition at line 465 of file des.c.

References `des3_set3key()`, and `des3_context::sk`.

Referenced by `des_self_test()`, `ssl_derive_keys()`, and `x509_des3_decrypt()`.

#### 13.121.2.8 **void des3\_set3key\_enc (des3\_context \* *ctx*, unsigned char *key*[24])**

Triple-DES key schedule (168-bit, encryption).

##### **Parameters:**

*ctx* 3DES context to be initialized

*key* 24-byte secret key

Definition at line 454 of file des.c.

References `des3_set3key()`, and `des3_context::sk`.

Referenced by `des_self_test()`, `main()`, and `ssl_derive_keys()`.

#### 13.121.2.9 **void des\_crypt\_cbc (des\_context \* *ctx*, int *mode*, int *length*, unsigned char *iv*[8], unsigned char \* *input*, unsigned char \* *output*)**

DES-CBC buffer encryption/decryption.

##### **Parameters:**

*ctx* DES context

*mode* DES\_ENCRYPT or DES\_DECRYPT

*length* length of the input data

*iv* initialization vector (updated after use)

*input* buffer holding the input data

*output* buffer holding the output data

Definition at line 505 of file des.c.

References `des_crypt_ecb()`, and `DES_ENCRYPT`.

Referenced by `des_self_test()`, and `main()`.

#### 13.121.2.10 **void des\_crypt\_ecb (des\_context \* *ctx*, unsigned char *input*[8], unsigned char *output*[8])**

DES-ECB block encryption/decryption.

##### **Parameters:**

*ctx* DES context

*input* 64-bit input block

*output* 64-bit output block

Definition at line 476 of file des.c.

References `DES_FP`, `DES_IP`, `DES_ROUND`, `GET_ULONG_BE`, `PUT_ULONG_BE`, and `des_context::sk`.

Referenced by `des_crypt_cbc()`, and `des_self_test()`.

### 13.121.2.11 `int des_self_test (int verbose)`

Definition at line 694 of file des.c.

References `buf`, `des3_crypt_cbc()`, `des3_crypt_ecb()`, `des3_set2key_dec()`, `des3_set2key_enc()`, `des3_set3key_dec()`, `des3_set3key_enc()`, `des3_test_buf`, `des3_test_cbc_dec`, `des3_test_cbc_enc`, `des3_test_ecb_dec`, `des3_test_ecb_enc`, `des3_test_iv`, `des3_test_keys`, `des_crypt_cbc()`, `des_crypt_ecb()`, `DES_DECRYPT`, `des_setkey_dec()`, `des_setkey_enc()`, and `prv`.

Referenced by `main()`.

### 13.121.2.12 `static void des_setkey (unsigned long SK[32], unsigned char key[8]) [static]`

Definition at line 288 of file des.c.

References `GET_ULONG_BE`, `LHs`, and `RHs`.

Referenced by `des3_set2key()`, `des3_set3key()`, `des_setkey_dec()`, and `des_setkey_enc()`.

### 13.121.2.13 `void des_setkey_dec (des_context * ctx, unsigned char key[8])`

DES key schedule (56-bit, decryption).

#### Parameters:

*ctx* DES context to be initialized

*key* 8-byte secret key

Definition at line 368 of file des.c.

References `des_setkey()`, `des_context::sk`, and `SWAP`.

Referenced by `des_self_test()`.

### 13.121.2.14 `void des_setkey_enc (des_context * ctx, unsigned char key[8])`

DES key schedule (56-bit, encryption).

#### Parameters:

*ctx* DES context to be initialized

*key* 8-byte secret key

Definition at line 360 of file des.c.

References `des_setkey()`, and `des_context::sk`.

Referenced by `des_self_test()`, and `main()`.

## 13.121.3 Variable Documentation

### 13.121.3.1 `const unsigned char des3_test_buf[8] [static]`

#### Initial value:

```
{
    0x4E, 0x6F, 0x77, 0x20, 0x69, 0x73, 0x20, 0x74
}
```

Definition at line 658 of file des.c.

Referenced by des\_self\_test().

### 13.121.3.2 `const unsigned char des3_test_cbc_dec[3][8] [static]`

**Initial value:**

```
{
    { 0x12, 0x9F, 0x40, 0xB9, 0xD2, 0x00, 0x56, 0xB3 },
    { 0x47, 0x0E, 0xFC, 0x9A, 0x6B, 0x8E, 0xE3, 0x93 },
    { 0xC5, 0xCE, 0xCF, 0x63, 0xEC, 0xEC, 0x51, 0x4C }
}
```

Definition at line 677 of file des.c.

Referenced by des\_self\_test().

### 13.121.3.3 `const unsigned char des3_test_cbc_enc[3][8] [static]`

**Initial value:**

```
{
    { 0x54, 0xF1, 0x5A, 0xF6, 0xEB, 0xE3, 0xA4, 0xB4 },
    { 0x35, 0x76, 0x11, 0x56, 0x5F, 0xA1, 0x8E, 0x4D },
    { 0xCB, 0x19, 0x1F, 0x85, 0xD1, 0xED, 0x84, 0x39 }
}
```

Definition at line 684 of file des.c.

Referenced by des\_self\_test().

### 13.121.3.4 `const unsigned char des3_test_ecb_dec[3][8] [static]`

**Initial value:**

```
{
    { 0xCD, 0xD6, 0x4F, 0x2F, 0x94, 0x27, 0xC1, 0x5D },
    { 0x69, 0x96, 0xC8, 0xFA, 0x47, 0xA2, 0xAB, 0xEB },
    { 0x83, 0x25, 0x39, 0x76, 0x44, 0x09, 0x1A, 0x0A }
}
```

Definition at line 663 of file des.c.

Referenced by des\_self\_test().

### 13.121.3.5 `const unsigned char des3_test_ecb_enc[3][8] [static]`

**Initial value:**



```
{
    { 0x6A, 0x2A, 0x19, 0xF4, 0x1E, 0xCA, 0x85, 0x4B },
    { 0x03, 0xE6, 0x9F, 0x5B, 0xFA, 0x58, 0xEB, 0x42 },
    { 0xDD, 0x17, 0xE8, 0xB8, 0xB4, 0x37, 0xD2, 0x32 }
}
```

Definition at line 670 of file des.c.

Referenced by des\_self\_test().

### 13.121.3.6 const unsigned char des3\_test\_iv[8] [static]

Initial value:

```
{
    0x12, 0x34, 0x56, 0x78, 0x90, 0xAB, 0xCD, 0xEF,
}
```

Definition at line 653 of file des.c.

Referenced by des\_self\_test().

### 13.121.3.7 const unsigned char des3\_test\_keys[24] [static]

Initial value:

```
{
    0x01, 0x23, 0x45, 0x67, 0x89, 0xAB, 0xCD, 0xEF,
    0x23, 0x45, 0x67, 0x89, 0xAB, 0xCD, 0xEF, 0x01,
    0x45, 0x67, 0x89, 0xAB, 0xCD, 0xEF, 0x01, 0x23
}
```

Definition at line 646 of file des.c.

Referenced by des\_self\_test().

### 13.121.3.8 const unsigned long LHs[16] [static]

Initial value:

```
{
    0x00000000, 0x00000001, 0x00000100, 0x00000101,
    0x00010000, 0x00010001, 0x00010100, 0x00010101,
    0x01000000, 0x01000001, 0x01000100, 0x01000101,
    0x01010000, 0x01010001, 0x01010100, 0x01010101
}
```

Definition at line 224 of file des.c.

Referenced by des\_setkey().

### 13.121.3.9 const unsigned long RHs[16] [static]

Initial value:

```
{
    0x00000000, 0x01000000, 0x00010000, 0x01010000,
    0x00000100, 0x01000100, 0x00010100, 0x01010100,
    0x00000001, 0x01000001, 0x00010001, 0x01010001,
    0x00000101, 0x01000101, 0x00010101, 0x01010101,
}
```

Definition at line 232 of file des.c.

Referenced by des\_setkey().

### 13.121.3.10 const unsigned long SB1[64] [static]

Initial value:

```
{
    0x01010400, 0x00000000, 0x00010000, 0x01010404,
    0x01010004, 0x00010404, 0x00000004, 0x00010000,
    0x00000400, 0x01010400, 0x01010404, 0x00000400,
    0x01000404, 0x01010004, 0x01000000, 0x00000004,
    0x00000404, 0x01000400, 0x01000400, 0x00010400,
    0x00010400, 0x01010000, 0x01010000, 0x01000404,
    0x00010004, 0x01000004, 0x01000004, 0x00010004,
    0x00000000, 0x00000404, 0x00010404, 0x01000000,
    0x00010000, 0x01010404, 0x00000004, 0x01010000,
    0x01010400, 0x01000000, 0x01000000, 0x00000400,
    0x01010004, 0x00010000, 0x00010400, 0x01000004,
    0x00000400, 0x00000004, 0x01000404, 0x00010404,
    0x01010404, 0x00010004, 0x01010000, 0x01000404,
    0x01000004, 0x00000404, 0x00010404, 0x01010400,
    0x00000404, 0x01000400, 0x01000400, 0x00000000,
    0x00010004, 0x00010400, 0x00000000, 0x01010004
}
```

Definition at line 61 of file des.c.

### 13.121.3.11 const unsigned long SB2[64] [static]

Initial value:

```
{
    0x80108020, 0x80008000, 0x00008000, 0x00108020,
    0x00100000, 0x00000020, 0x80100020, 0x80008020,
    0x80000020, 0x80108020, 0x80108000, 0x80000000,
    0x80008000, 0x00100000, 0x00000020, 0x80100020,
    0x00108000, 0x00100020, 0x80008020, 0x00000000,
    0x80000000, 0x00008000, 0x00108020, 0x80100000,
    0x00100020, 0x80000020, 0x00000000, 0x00108000,
    0x00008020, 0x80108000, 0x80100000, 0x00008020,
    0x00000000, 0x00108020, 0x80100020, 0x00100000,
    0x80008020, 0x80100000, 0x80108000, 0x00008000,
    0x80100000, 0x80008000, 0x00000020, 0x80108020,
    0x00108020, 0x00000020, 0x00008000, 0x80000000,
    0x00008020, 0x80108000, 0x00100000, 0x80000020,
    0x00100020, 0x80008020, 0x80000020, 0x00100020,
    0x00108000, 0x00000000, 0x80008000, 0x00008020,
    0x80000000, 0x80100020, 0x80108020, 0x00108000
}
```

Definition at line 81 of file des.c.

### 13.121.3.12 const unsigned long SB3[64] [static]

Initial value:

```
{
    0x00000208, 0x08020200, 0x00000000, 0x08020008,
    0x08000200, 0x00000000, 0x00020208, 0x08000200,
    0x00020008, 0x08000008, 0x08000008, 0x00020000,
    0x08020208, 0x00020008, 0x08020000, 0x00000208,
    0x08000000, 0x00000008, 0x08020200, 0x00000200,
    0x00020200, 0x08020000, 0x08020008, 0x00020208,
    0x08000208, 0x00020200, 0x00020000, 0x08000208,
    0x00000008, 0x08020208, 0x00000200, 0x08000000,
    0x08020200, 0x08000000, 0x00020008, 0x00000208,
    0x00020000, 0x08020200, 0x08000200, 0x00000000,
    0x00000200, 0x00020008, 0x08020208, 0x08000200,
    0x08000008, 0x00000200, 0x00000000, 0x08020008,
    0x08000208, 0x00020000, 0x08000000, 0x08020208,
    0x00000008, 0x00020208, 0x00020200, 0x08000008,
    0x08020000, 0x08000208, 0x00000208, 0x08020000,
    0x00020208, 0x00000008, 0x08020008, 0x00020200
}
```

Definition at line 101 of file des.c.

### 13.121.3.13 const unsigned long SB4[64] [static]

Initial value:

```
{
    0x00802001, 0x00002081, 0x00002081, 0x00000080,
    0x00802080, 0x00800081, 0x00800001, 0x00002001,
    0x00000000, 0x00802000, 0x00802000, 0x00802081,
    0x00000081, 0x00000000, 0x00800080, 0x00800001,
    0x00000001, 0x00002000, 0x00800000, 0x00802001,
    0x00000080, 0x00800000, 0x00002001, 0x00002080,
    0x00800081, 0x00000001, 0x00002080, 0x00800080,
    0x00002000, 0x00802080, 0x00802081, 0x00000081,
    0x00800080, 0x00800001, 0x00802000, 0x00802081,
    0x00000081, 0x00000000, 0x00000000, 0x00802000,
    0x00002080, 0x00800080, 0x00800081, 0x00000001,
    0x00802001, 0x00002081, 0x00002081, 0x00000080,
    0x00802081, 0x00000081, 0x00000001, 0x00002000,
    0x00800001, 0x00002001, 0x00802080, 0x00800081,
    0x00002001, 0x00002080, 0x00800000, 0x00802001,
    0x00000080, 0x00800000, 0x00002000, 0x00802080
}
```

Definition at line 121 of file des.c.

### 13.121.3.14 const unsigned long SB5[64] [static]

Initial value:

```
{
    0x00000100, 0x02080100, 0x02080000, 0x42000100,
    0x00080000, 0x00000100, 0x40000000, 0x02080000,
    0x40080100, 0x00080000, 0x02000100, 0x40080100,
    0x42000100, 0x42080000, 0x00080100, 0x40000000,
}
```

```

0x02000000, 0x40080000, 0x40080000, 0x00000000,
0x40000100, 0x42080100, 0x42080100, 0x02000100,
0x42080000, 0x40000100, 0x00000000, 0x42000000,
0x02080100, 0x02000000, 0x42000000, 0x00080100,
0x00080000, 0x42000100, 0x00000100, 0x02000000,
0x40000000, 0x02080000, 0x42000100, 0x40080100,
0x02000100, 0x40000000, 0x42080000, 0x02080100,
0x40080100, 0x00000100, 0x02000000, 0x42080000,
0x42080100, 0x00080100, 0x42000000, 0x42080100,
0x02080000, 0x00000000, 0x40080000, 0x42000000,
0x00080100, 0x02000100, 0x40000100, 0x00080000,
0x00000000, 0x40080000, 0x02080100, 0x40000100
}

```

Definition at line 141 of file des.c.

### 13.121.3.15 const unsigned long SB6[64] [static]

Initial value:

```

{
    0x20000010, 0x20400000, 0x00004000, 0x20404010,
    0x20400000, 0x00000010, 0x20404010, 0x00400000,
    0x20004000, 0x00404010, 0x00400000, 0x20000010,
    0x00400010, 0x20004000, 0x20000000, 0x00004010,
    0x00000000, 0x00400010, 0x20004010, 0x00004000,
    0x00404000, 0x20004010, 0x00000010, 0x20400010,
    0x20400010, 0x00000000, 0x00404010, 0x20404000,
    0x00004010, 0x00404000, 0x20404000, 0x20000000,
    0x20004000, 0x00000010, 0x20400010, 0x00404000,
    0x20404010, 0x00400000, 0x00000010, 0x20000010,
    0x00400000, 0x20004000, 0x20000000, 0x00004010,
    0x20000010, 0x20404010, 0x00404000, 0x20400000,
    0x00404010, 0x20404000, 0x00000000, 0x20400010,
    0x00000010, 0x00004000, 0x20400000, 0x00404010,
    0x00004000, 0x00400010, 0x20004010, 0x00000000,
    0x20404000, 0x20000000, 0x00400010, 0x20004010
}

```

Definition at line 161 of file des.c.

### 13.121.3.16 const unsigned long SB7[64] [static]

Initial value:

```

{
    0x00200000, 0x04200002, 0x04000802, 0x00000000,
    0x00000800, 0x04000802, 0x00200802, 0x04200800,
    0x04200802, 0x00200000, 0x00000000, 0x04000002,
    0x00000002, 0x04000000, 0x04200002, 0x00000802,
    0x04000800, 0x00200802, 0x00200002, 0x04000800,
    0x04000002, 0x04200000, 0x04200800, 0x00200002,
    0x04200000, 0x00000800, 0x00000802, 0x04200802,
    0x00200800, 0x00000002, 0x04000000, 0x00200800,
    0x04000000, 0x00200800, 0x00200000, 0x04000802,
    0x04000802, 0x04200002, 0x04200002, 0x00000002,
    0x00200002, 0x04000000, 0x04000800, 0x00200000,
    0x04200800, 0x00000802, 0x00200802, 0x04200800,
    0x00000802, 0x04000002, 0x04200802, 0x04200000,
    0x00200800, 0x00000000, 0x00000002, 0x04200802,
}

```

```
    0x00000000, 0x00200802, 0x04200000, 0x00000800,  
    0x04000002, 0x04000800, 0x00000800, 0x00200002  
}
```

Definition at line 181 of file des.c.

### 13.121.3.17 const unsigned long SB8[64] [static]

Initial value:

```
{  
    0x10001040, 0x00001000, 0x00040000, 0x10041040,  
    0x10000000, 0x10001040, 0x00000040, 0x10000000,  
    0x00040040, 0x10040000, 0x10041040, 0x00041000,  
    0x10041000, 0x00041040, 0x00001000, 0x00000040,  
    0x10040000, 0x10000040, 0x10001000, 0x00001040,  
    0x00041000, 0x00040040, 0x10040040, 0x10041000,  
    0x00001040, 0x00000000, 0x00000000, 0x10040040,  
    0x10000040, 0x10001000, 0x00041040, 0x00040000,  
    0x00041040, 0x00040000, 0x10041000, 0x00001000,  
    0x00000040, 0x10040040, 0x00001000, 0x00041040,  
    0x10001000, 0x00000040, 0x10000040, 0x10040000,  
    0x10040040, 0x10000000, 0x00040000, 0x10001040,  
    0x00000000, 0x10041040, 0x00040040, 0x10000040,  
    0x10040000, 0x10001000, 0x10001040, 0x00000000,  
    0x10041040, 0x00041000, 0x00041000, 0x00001040,  
    0x00001040, 0x00040040, 0x10000000, 0x10041000  
}
```

Definition at line 201 of file des.c.

## 13.122 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/library/dhm.c File Reference

```
#include "xyssl/config.h"
#include "xyssl/dhm.h"
#include <string.h>
```

### Defines

- #define [DHM\\_MPI\\_EXPORT](#)(X, n)

### Functions

- static [int dhm\\_read\\_bignum](#) ([mpi](#) \*X, unsigned char \*\*p, unsigned char \*end)
- [int dhm\\_read\\_params](#) ([dhm\\_context](#) \*ctx, unsigned char \*\*p, unsigned char \*end)  
*Parse the ServerKeyExchange parameters.*
- [int dhm\\_make\\_params](#) ([dhm\\_context](#) \*ctx, [int](#) x\_size, unsigned char \*output, [int](#) \*olen, [int](#)(\*f\_rng)(void \*), void \*p\_rng)  
*Setup and write the ServerKeyExchange parameters.*
- [int dhm\\_read\\_public](#) ([dhm\\_context](#) \*ctx, unsigned char \*input, [int](#) ilen)  
*Import the peer's public value  $G^Y$ .*
- [int dhm\\_make\\_public](#) ([dhm\\_context](#) \*ctx, [int](#) x\_size, unsigned char \*output, [int](#) olen, [int](#)(\*f\_rng)(void \*), void \*p\_rng)  
*Create own private value X and export  $G^X$ .*
- [int dhm\\_calc\\_secret](#) ([dhm\\_context](#) \*ctx, unsigned char \*output, [int](#) \*olen)  
*Derive and export the shared secret  $(G^Y)^X \bmod P$ .*
- void [dhm\\_free](#) ([dhm\\_context](#) \*ctx)
- [int dhm\\_self\\_test](#) ([int](#) verbose)  
*Checkup routine.*

### 13.122.1 Define Documentation

#### 13.122.1.1 #define DHM\_MPI\_EXPORT(X, n)

##### Value:

```
MPI_CHK( mpi_write_binary( X, p + 2, n ) ); \
    *p++ = (unsigned char)( n >> 8 ); \
    *p++ = (unsigned char)( n ); p += n;
```

Referenced by [dhm\\_make\\_params\(\)](#).

## 13.122.2 Function Documentation

### 13.122.2.1 `int dhm_calc_secret(dhm_context * ctx, unsigned char * output, int * olen)`

Derive and export the shared secret  $(G^Y)^X \bmod P$ .

#### Parameters:

*ctx* DHM context  
*output* destination buffer  
*olen* number of chars written

#### Returns:

0 if successful, or an XYSSL\_ERR\_DHM\_XXX error code

Definition at line 208 of file dhm.c.

References `dhm_context::GY`, `dhm_context::K`, `MPI_CHK`, `mpi_exp_mod()`, `mpi_size()`, `mpi_write_binary()`, `dhm_context::P`, `dhm_context::RP`, `dhm_context::X`, `XYSSL_ERR_DHM_BAD_INPUT_DATA`, and `XYSSL_ERR_DHM_CALC_SECRET_FAILED`.

Referenced by `main()`, `ssl_parse_client_key_exchange()`, and `ssl_write_client_key_exchange()`.

### 13.122.2.2 `void dhm_free(dhm_context * ctx)`

Definition at line 234 of file dhm.c.

References `dhm_context::G`, `dhm_context::GX`, `dhm_context::GY`, `dhm_context::K`, `mpi_free()`, `dhm_context::P`, `dhm_context::RP`, and `dhm_context::X`.

Referenced by `main()`, and `ssl_free()`.

### 13.122.2.3 `int dhm_make_params(dhm_context * ctx, int s_size, unsigned char * output, int * olen, int(*) (void *) f_rng, void * p_rng)`

Setup and write the ServerKeyExchange parameters.

#### Parameters:

*ctx* DHM context  
*x\_size* private value size in bits  
*output* destination buffer  
*olen* number of chars written  
*f\_rng* RNG function  
*p\_rng* RNG parameter

#### Note:

This function assumes that `ctx->P` and `ctx->G` have already been properly set (for example using `mpi_read_string` or `mpi_read_binary`).

#### Returns:

0 if successful, or an XYSSL\_ERR\_DHM\_XXX error code

Definition at line 93 of file dhm.c.

References DHM\_MPI\_EXPORT, dhm\_context::G, dhm\_context::GX, dhm\_context::len, MPI\_CHK, mpi\_cmp\_mpi(), mpi\_exp\_mod(), mpi\_grow(), mpi\_lset(), mpi\_shift\_r(), mpi\_size(), dhm\_context::P, mpi::p, dhm\_context::RP, dhm\_context::X, and XYSSL\_ERR\_DHM\_MAKE\_PARAMS\_FAILED.

Referenced by main(), and ssl\_write\_server\_key\_exchange().

**13.122.2.4** `int dhm_make_public(dhm_context * ctx, int s_size, unsigned char * output, int olen, int(*) (void *) f_rng, void * p_rng)`

Create own private value X and export  $G^X$ .

**Parameters:**

*ctx* DHM context  
*x\_size* private value size in bits  
*output* destination buffer  
*olen* must be equal to `ctx->P.len`  
*f\_rng* RNG function  
*p\_rng* RNG parameter

**Returns:**

0 if successful, or an XYSSL\_ERR\_DHM\_XXX error code

Definition at line 167 of file dhm.c.

References dhm\_context::G, dhm\_context::GX, dhm\_context::len, MPI\_CHK, mpi\_cmp\_mpi(), mpi\_exp\_mod(), mpi\_grow(), mpi\_lset(), mpi\_shift\_r(), mpi\_write\_binary(), dhm\_context::P, mpi::p, dhm\_context::RP, dhm\_context::X, XYSSL\_ERR\_DHM\_BAD\_INPUT\_DATA, and XYSSL\_ERR\_DHM\_MAKE\_PUBLIC\_FAILED.

Referenced by main(), and ssl\_write\_client\_key\_exchange().

**13.122.2.5** `static int dhm_read_bignum(mpi * X, unsigned char ** p, unsigned char * end) [static]`

Definition at line 37 of file dhm.c.

References mpi\_read\_binary(), XYSSL\_ERR\_DHM\_BAD\_INPUT\_DATA, and XYSSL\_ERR\_DHM\_READ\_PARAMS\_FAILED.

Referenced by dhm\_read\_params().

**13.122.2.6** `int dhm_read_params(dhm_context * ctx, unsigned char ** p, unsigned char * end)`

Parse the ServerKeyExchange parameters.

**Parameters:**

*ctx* DHM context  
*p* &(start of input buffer)  
*end* end of buffer



**Returns:**

0 if successful, or an XYSSL\_ERR\_DHM\_XXX error code

Definition at line 63 of file dhm.c.

References dhm\_read\_bignum(), dhm\_context::G, dhm\_context::GY, dhm\_context::len, mpi\_size(), dhm\_context::P, and XYSSL\_ERR\_DHM\_BAD\_INPUT\_DATA.

Referenced by main(), and ssl\_parse\_server\_key\_exchange().

**13.122.2.7 int dhm\_read\_public (dhm\_context \* *ctx*, unsigned char \* *input*, int *ilen*)**

Import the peer's public value  $G^Y$ .

**Parameters:**

*ctx* DHM context

*input* input buffer

*ilen* size of buffer

**Returns:**

0 if successful, or an XYSSL\_ERR\_DHM\_XXX error code

Definition at line 150 of file dhm.c.

References dhm\_context::GY, dhm\_context::len, mpi\_read\_binary(), XYSSL\_ERR\_DHM\_BAD\_INPUT\_DATA, and XYSSL\_ERR\_DHM\_READ\_PUBLIC\_FAILED.

Referenced by main(), and ssl\_parse\_client\_key\_exchange().

**13.122.2.8 int dhm\_self\_test (int *verbose*)**

Checkup routine.

**Returns:**

0 if successful, or 1 if the test failed

Definition at line 246 of file dhm.c.

## 13.123 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/library/havege.c File Reference

```
#include <string.h>
#include <time.h>
#include "xyssl/config.h"
#include "xyssl/havege.h"
#include "xyssl/timing.h"
```

### Defines

- #define [SWAP\(X, Y\)](#) { int \*T = X; X = Y; Y = T; }
- #define [TST1\\_ENTER](#) if( PTEST & 1 ) { PTEST ^= 3; PTEST >>= 1;
- #define [TST2\\_ENTER](#) if( PTEST & 1 ) { PTEST ^= 3; PTEST >>= 1;
- #define [TST1\\_LEAVE](#) U1++; }
- #define [TST2\\_LEAVE](#) U2++; }
- #define [ONE\\_ITERATION](#)

### Functions

- static void [havege\\_fill](#) (havege\_state \*hs)
- void [havege\\_init](#) (havege\_state \*hs)  
*HAVEGE initialization.*
- int [havege\\_rand](#) (void \*p\_rng)  
*HAVEGE rand function.*

### 13.123.1 Define Documentation

#### 13.123.1.1 #define ONE\_ITERATION

Definition at line 60 of file havege.c.

Referenced by [havege\\_fill\(\)](#).

#### 13.123.1.2 #define SWAP(X, Y) { int \*T = X; X = Y; Y = T; }

Definition at line 52 of file havege.c.

#### 13.123.1.3 #define TST1\_ENTER if( PTEST & 1 ) { PTEST ^= 3; PTEST >>= 1;

Definition at line 54 of file havege.c.

#### 13.123.1.4 `#define TST1_LEAVE U1++; }`

Definition at line 57 of file havege.c.

#### 13.123.1.5 `#define TST2_ENTER if( PTEST & 1 ) { PTEST ^= 3; PTEST >>= 1;`

Definition at line 55 of file havege.c.

#### 13.123.1.6 `#define TST2_LEAVE U2++; }`

Definition at line 58 of file havege.c.

### 13.123.2 Function Documentation

#### 13.123.2.1 `static void havege_fill (havege_state * hs) [static]`

Definition at line 154 of file havege.c.

References `COLLECT_SIZE`, `havege_state::offset`, `ONE_ITERATION`, `havege_state::PT1`, `havege_state::PT2`, and `havege_state::WALK`.

Referenced by `havege_init()`, and `havege_rand()`.

#### 13.123.2.2 `void havege_init (havege_state * hs)`

HAVEGE initialization.

##### Parameters:

*hs* HAVEGE state to be initialized

Definition at line 188 of file havege.c.

References `havege_fill()`.

Referenced by `generate_AES_key()`, `generate_RSA_keys_ciphertext()`, `generate_RSA_keys_plaintext()`, `initiate_migration_process()`, `main()`, and `ssl_test()`.

#### 13.123.2.3 `int havege_rand (void * p_rng)`

HAVEGE rand function.

##### Parameters:

*rng\_st* points to an HAVEGE state

##### Returns:

A random int

Definition at line 198 of file havege.c.

References `COLLECT_SIZE`, `havege_fill()`, `havege_state::offset`, and `havege_state::pool`.

Referenced by `generate_AES_key()`, `generate_RSA_keys_ciphertext()`, `generate_RSA_keys_plaintext()`, `initiate_migration_process()`, `main()`, and `ssl_test()`.

### **13.124    /home/dko/Projects/mobilec/tags/MobileC- v1.10.10/src/security/xyssl-0.9/library/md2.c File Reference**

```
#include "xyssl/config.h"
```

## 13.125 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/library/md4.c File Reference

```
#include "xyssl/config.h"
```

## 13.126 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/library/md5.c File Reference

```
#include "xyssl/config.h"
#include "xyssl/md5.h"
#include <string.h>
#include <stdio.h>
```

### Defines

- `#define S(x, n) ((x << n) | ((x & 0xFFFFFFFF) >> (32 - n)))`
- `#define P(a, b, c, d, k, s, t)`
- `#define F(x, y, z) (z ^ (x & (y ^ z)))`
- `#define F(x, y, z) (y ^ (z & (x ^ y)))`
- `#define F(x, y, z) (x ^ y ^ z)`
- `#define F(x, y, z) (y ^ (x | ~z))`

### Functions

- void `md5_starts` (`md5_context` \*ctx)  
*MD5 context setup.*
- static void `md5_process` (`md5_context` \*ctx, unsigned char data[64])
- void `md5_update` (`md5_context` \*ctx, unsigned char \*input, `int` ilen)  
*MD5 process buffer.*
- void `md5_finish` (`md5_context` \*ctx, unsigned char output[16])  
*MD5 final digest.*
- void `md5` (unsigned char \*input, `int` ilen, unsigned char output[16])  
*Output = MD5( input buffer ).*
- `int` `md5_file` (char \*path, unsigned char output[16])  
*Output = MD5( file contents ).*
- void `md5_hmac_starts` (`md5_context` \*ctx, unsigned char \*key, `int` keylen)  
*MD5 HMAC context setup.*
- void `md5_hmac_update` (`md5_context` \*ctx, unsigned char \*input, `int` ilen)  
*MD5 HMAC process buffer.*
- void `md5_hmac_finish` (`md5_context` \*ctx, unsigned char output[16])  
*MD5 HMAC final digest.*
- void `md5_hmac` (unsigned char \*key, `int` keylen, unsigned char \*input, `int` ilen, unsigned char output[16])  
*Output = HMAC-MD5( hmac key, input buffer ).*

- `int md5_self_test (int verbose)`  
*Checkup routine.*

## Variables

- static const unsigned char `md5_padding` [64]
- static unsigned char `md5_test_buf` [7][81]
- static const `int md5_test_buflen` [7]
- static const unsigned char `md5_test_sum` [7][16]
- static unsigned char `md5_hmac_test_key` [7][26]
- static const `int md5_hmac_test_keylen` [7]
- static unsigned char `md5_hmac_test_buf` [7][74]
- static const `int md5_hmac_test_buflen` [7]
- static const unsigned char `md5_hmac_test_sum` [7][16]

## 13.126.1 Define Documentation

**13.126.1.1** `#define F(x, y, z) (y ^ (x | ~z))`

**13.126.1.2** `#define F(x, y, z) (x ^ y ^ z)`

**13.126.1.3** `#define F(x, y, z) (y ^ (z & (x ^ y)))`

**13.126.1.4** `#define F(x, y, z) (z ^ (x & (y ^ z)))`

Referenced by `sha2_process()`, and `sha4_process()`.

**13.126.1.5** `#define P(a, b, c, d, k, s, t)`

**Value:**

```
{
    a += F(b, c, d) + X[k] + t; a = S(a, s) + b;
}
```

Referenced by `main()`, `md5_process()`, `rsa_decryption()`, `sha1_process()`, `sha2_process()`, and `sha4_process()`.

**13.126.1.6** `#define S(x, n) ((x << n) | ((x & 0xFFFFFFFF) >> (32 - n)))`

## 13.126.2 Function Documentation

**13.126.2.1** `void md5 (unsigned char *input, int ilen, unsigned char output[16])`

Output = MD5( input buffer ).

**Parameters:**

*input* buffer holding the data

*ilen* length of the input data

*output* MD5 checksum result

Definition at line 278 of file md5.c.

References md5\_finish(), md5\_starts(), and md5\_update().

Referenced by initiate\_migration\_process(), main(), md5\_hmac\_starts(), md5\_self\_test(), reply\_migration\_process(), ssl\_calc\_verify(), ssl\_derive\_keys(), ssl\_mac\_md5(), ssl\_parse\_finished(), ssl\_parse\_server\_key\_exchange(), ssl\_write\_finished(), ssl\_write\_server\_key\_exchange(), and x509\_hash().

### 13.126.2.2 int md5\_file (char \* *path*, unsigned char *output*[16])

Output = MD5( file contents ).

#### Parameters:

*path* input file name

*output* MD5 checksum result

#### Returns:

0 if successful, 1 if fopen failed, or 2 if fread failed

Definition at line 292 of file md5.c.

References buf, f, md5\_finish(), md5\_starts(), and md5\_update().

Referenced by md5\_wrapper().

### 13.126.2.3 void md5\_finish (md5\_context \* *ctx*, unsigned char *output*[16])

MD5 final digest.

#### Parameters:

*ctx* MD5 context

*output* MD5 checksum result

Definition at line 250 of file md5.c.

References md5\_padding, md5\_update(), PUT\_ULONG\_LE, md5\_context::state, and md5\_context::total.

Referenced by md5(), md5\_file(), md5\_hmac\_finish(), ssl\_calc\_finished(), ssl\_calc\_verify(), ssl\_derive\_keys(), ssl\_mac\_md5(), ssl\_parse\_server\_key\_exchange(), ssl\_write\_server\_key\_exchange(), and x509\_des3\_decrypt().

### 13.126.2.4 void md5\_hmac (unsigned char \* *key*, int *keylen*, unsigned char \* *input*, int *ilen*, unsigned char *output*[16])

Output = HMAC-MD5( hmac key, input buffer ).

#### Parameters:

*key* HMAC secret key



*keylen* length of the HMAC key  
*input* buffer holding the data  
*ilen* length of the input data  
*output* HMAC-MD5 result

Definition at line 378 of file md5.c.

References md5\_hmac\_finish(), md5\_hmac\_starts(), and md5\_hmac\_update().

Referenced by ssl\_decrypt\_buf(), ssl\_encrypt\_buf(), and tls1\_prf().

#### 13.126.2.5 void md5\_hmac\_finish (md5\_context \* ctx, unsigned char output[16])

MD5 HMAC final digest.

##### Parameters:

*ctx* HMAC context  
*output* MD5 HMAC checksum result

Definition at line 362 of file md5.c.

References md5\_finish(), md5\_starts(), md5\_update(), and md5\_context::opad.

Referenced by md5\_hmac(), and md5\_self\_test().

#### 13.126.2.6 void md5\_hmac\_starts (md5\_context \* ctx, unsigned char \* key, int keylen)

MD5 HMAC context setup.

##### Parameters:

*ctx* HMAC context to be initialized  
*key* HMAC secret key  
*keylen* length of the HMAC key

Definition at line 324 of file md5.c.

References md5\_context::ipad, md5(), md5\_starts(), md5\_update(), and md5\_context::opad.

Referenced by md5\_hmac(), and md5\_self\_test().

#### 13.126.2.7 void md5\_hmac\_update (md5\_context \* ctx, unsigned char \* input, int ilen)

MD5 HMAC process buffer.

##### Parameters:

*ctx* HMAC context  
*input* buffer holding the data  
*ilen* length of the input data

Definition at line 354 of file md5.c.

References md5\_update().

Referenced by md5\_hmac(), and md5\_self\_test().

**13.126.2.8 static void md5\_process (md5\_context \* *ctx*, unsigned char *data*[64]) [static]**

Definition at line 72 of file md5.c.

References GET\_ULONG\_LE, P, and md5\_context::state.

Referenced by md5\_update().

**13.126.2.9 int md5\_self\_test (int *verbose*)**

Checksum routine.

**Returns:**

0 if successful, or 1 if the test failed

Definition at line 495 of file md5.c.

References buf, md5(), md5\_hmac\_finish(), md5\_hmac\_starts(), md5\_hmac\_test\_buf, md5\_hmac\_test\_buflen, md5\_hmac\_test\_key, md5\_hmac\_test\_keylen, md5\_hmac\_test\_sum, md5\_hmac\_update(), md5\_test\_buf, md5\_test\_buflen, and md5\_test\_sum.

Referenced by main().

**13.126.2.10 void md5\_starts (md5\_context \* *ctx*)**

MD5 context setup.

**Parameters:**

*ctx* context to be initialized

Definition at line 61 of file md5.c.

References md5\_context::state, and md5\_context::total.

Referenced by md5(), md5\_file(), md5\_hmac\_finish(), md5\_hmac\_starts(), ssl\_calc\_finished(), ssl\_calc\_verify(), ssl\_derive\_keys(), ssl\_init(), ssl\_mac\_md5(), ssl\_parse\_server\_key\_exchange(), ssl\_write\_server\_key\_exchange(), and x509\_des3\_decrypt().

**13.126.2.11 void md5\_update (md5\_context \* *ctx*, unsigned char \* *input*, int *ilen*)**

MD5 process buffer.

**Parameters:**

*ctx* MD5 context

*input* buffer holding the data

*ilen* length of the input data

Definition at line 198 of file md5.c.

References md5\_context::buffer, md5\_process(), and md5\_context::total.

Referenced by md5(), md5\_file(), md5\_finish(), md5\_hmac\_finish(), md5\_hmac\_starts(), md5\_hmac\_update(), ssl\_calc\_finished(), ssl\_calc\_verify(), ssl\_derive\_keys(), ssl\_mac\_md5(), ssl\_parse\_client\_hello(), ssl\_parse\_server\_key\_exchange(), ssl\_read\_record(), ssl\_write\_record(), ssl\_write\_server\_key\_exchange(), and x509\_des3\_decrypt().



**13.126.3.4 const int md5\_hmac\_test\_keylen[7] [static]****Initial value:**

```
{
    16, 4, 16, 25, 16, 80, 80
}
```

Definition at line 444 of file md5.c.

Referenced by md5\_self\_test().

**13.126.3.5 const unsigned char md5\_hmac\_test\_sum[7][16] [static]****Initial value:**

```
{
    { 0x92, 0x94, 0x72, 0x7A, 0x36, 0x38, 0xBB, 0x1C,
      0x13, 0xF4, 0x8E, 0xF8, 0x15, 0x8B, 0xFC, 0x9D },
    { 0x75, 0x0C, 0x78, 0x3E, 0x6A, 0xB0, 0xB5, 0x03,
      0xEA, 0xA8, 0x6E, 0x31, 0x0A, 0x5D, 0xB7, 0x38 },
    { 0x56, 0xBE, 0x34, 0x52, 0x1D, 0x14, 0x4C, 0x88,
      0xDB, 0xB8, 0xC7, 0x33, 0xF0, 0xE8, 0xB3, 0xF6 },
    { 0x69, 0x7E, 0xAF, 0x0A, 0xCA, 0x3A, 0x3A, 0xEA,
      0x3A, 0x75, 0x16, 0x47, 0x46, 0xFF, 0xAA, 0x79 },
    { 0x56, 0x46, 0x1E, 0xF2, 0x34, 0x2E, 0xDC, 0x00,
      0xF9, 0xBA, 0xB9, 0x95 },
    { 0x6B, 0x1A, 0xB7, 0xFE, 0x4B, 0xD7, 0xBF, 0x8F,
      0x0B, 0x62, 0xE6, 0xCE, 0x61, 0xB9, 0xD0, 0xCD },
    { 0x6F, 0x63, 0x0F, 0xAD, 0x67, 0xCD, 0xA0, 0xEE,
      0x1F, 0xB1, 0xF5, 0x62, 0xDB, 0x3A, 0xA5, 0x3E }
}
```

Definition at line 474 of file md5.c.

Referenced by md5\_self\_test().

**13.126.3.6 const unsigned char md5\_padding[64] [static]****Initial value:**

```
{
    0x80, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
}
```

Definition at line 239 of file md5.c.

Referenced by md5\_finish().

**13.126.3.7 unsigned char md5\_test\_buf[7][81] [static]****Initial value:**

```
{
    { "" },
    { "a" },
    { "abc" },
    { "message digest" },
    { "abcdefghijklmnopqrstuvwxyz" },
    { "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789" },
    { "1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890" }
}
```

Definition at line 394 of file md5.c.

Referenced by md5\_self\_test().

### 13.126.3.8 const int md5\_test\_buflen[7] [static]

**Initial value:**

```
{
    0, 1, 3, 14, 26, 62, 80
}
```

Definition at line 406 of file md5.c.

Referenced by md5\_self\_test().

### 13.126.3.9 const unsigned char md5\_test\_sum[7][16] [static]

**Initial value:**

```
{
    { 0xD4, 0x1D, 0x8C, 0xD9, 0x8F, 0x00, 0xB2, 0x04,
      0xE9, 0x80, 0x09, 0x98, 0xEC, 0xF8, 0x42, 0x7E },
    { 0x0C, 0xC1, 0x75, 0xB9, 0xC0, 0xF1, 0xB6, 0xA8,
      0x31, 0xC3, 0x99, 0xE2, 0x69, 0x77, 0x26, 0x61 },
    { 0x90, 0x01, 0x50, 0x98, 0x3C, 0xD2, 0x4F, 0xB0,
      0xD6, 0x96, 0x3F, 0x7D, 0x28, 0xE1, 0x7F, 0x72 },
    { 0xF9, 0x6B, 0x69, 0x7D, 0x7C, 0xB7, 0x93, 0x8D,
      0x52, 0x5A, 0x2F, 0x31, 0xAA, 0xF1, 0x61, 0xD0 },
    { 0xC3, 0xFC, 0xD3, 0xD7, 0x61, 0x92, 0xE4, 0x00,
      0x7D, 0xFB, 0x49, 0x6C, 0xCA, 0x67, 0xE1, 0x3B },
    { 0xD1, 0x74, 0xAB, 0x98, 0xD2, 0x77, 0xD9, 0xF5,
      0xA5, 0x61, 0x1C, 0x2C, 0x9F, 0x41, 0x9D, 0x9F },
    { 0x57, 0xED, 0xF4, 0xA2, 0x2B, 0xE3, 0xC9, 0x55,
      0xAC, 0x49, 0xDA, 0x2E, 0x21, 0x07, 0xB6, 0x7A }
}
```

Definition at line 411 of file md5.c.

Referenced by md5\_self\_test().

## 13.127 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/library/net.c File Reference

```
#include "xyssl/config.h"
#include "xyssl/net.h"
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <sys/time.h>
#include <unistd.h>
#include <signal.h>
#include <fcntl.h>
#include <netdb.h>
#include <errno.h>
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <time.h>
```

### Functions

- static unsigned short [net\\_htons](#) (int port)
- [int net\\_connect](#) (int \*fd, char \*host, int port)  
*Initiate a TCP connection with host:port.*
- [int net\\_bind](#) (int \*fd, char \*bind\_ip, int port)  
*Create a listening socket on bind\_ip:port. If bind\_ip == NULL, all interfaces are binded.*
- static [int net\\_is\\_blocking](#) (void)
- [int net\\_accept](#) (int bind\_fd, int \*client\_fd, void \*client\_ip)  
*Accept a connection from a remote client.*
- [int net\\_set\\_block](#) (int fd)  
*Set the socket blocking.*
- [int net\\_set\\_nonblock](#) (int fd)  
*Set the socket non-blocking.*
- void [net\\_usleep](#) (unsigned long usec)  
*Portable usleep helper.*
- [int net\\_recv](#) (void \*ctx, unsigned char \*buf, int len)

*Read at most 'len' characters. len is updated to reflect the actual number of characters read.*

- `int net_send` (`void *ctx`, `unsigned char *buf`, `int len`)

*Write at most 'len' characters. len is updated to reflect the number of characters `_not_` written.*

- `void net_close` (`int fd`)

*Gracefully shutdown the connection.*

## 13.127.1 Function Documentation

### 13.127.1.1 `int net_accept` (`int bind_fd`, `int * client_fd`, `void * client_ip`)

Accept a connection from a remote client.

#### Returns:

0 if successful, `XYSSL_ERR_NET_ACCEPT_FAILED`, or `XYSSL_ERR_NET_WOULD_BLOCK` if `bind_fd` was set to non-blocking and `accept()` is blocking.

Definition at line 212 of file `net.c`.

References `int`, `net_is_blocking()`, `XYSSL_ERR_NET_ACCEPT_FAILED`, and `XYSSL_ERR_NET_TRY_AGAIN`.

Referenced by `main()`, and `ssl_test()`.

### 13.127.1.2 `int net_bind` (`int * fd`, `char * bind_ip`, `int port`)

Create a listening socket on `bind_ip:port`. If `bind_ip == NULL`, all interfaces are binded.

#### Returns:

0 if successful, or one of: `XYSSL_ERR_NET_SOCKET_FAILED`, `XYSSL_ERR_NET_BIND_FAILED`, `XYSSL_ERR_NET_LISTEN_FAILED`

Definition at line 126 of file `net.c`.

References `net_htons()`, `SOCKET_ERROR`, `XYSSL_ERR_NET_BIND_FAILED`, `XYSSL_ERR_NET_LISTEN_FAILED`, and `XYSSL_ERR_NET_SOCKET_FAILED`.

Referenced by `main()`, and `ssl_test()`.

### 13.127.1.3 `void net_close` (`int fd`)

Gracefully shutdown the connection.

Definition at line 338 of file `net.c`.

Referenced by `main()`, and `ssl_test()`.

### 13.127.1.4 `int net_connect` (`int * fd`, `char * host`, `int port`)

Initiate a TCP connection with `host:port`.

**Returns:**

0 if successful, or one of: XYSSL\_ERR\_NET\_SOCKET\_FAILED, XYSSL\_ERR\_NET\_UNKNOWN\_HOST, XYSSL\_ERR\_NET\_CONNECT\_FAILED

Definition at line 81 of file net.c.

References net\_htons(), SOCKET\_ERROR, XYSSL\_ERR\_NET\_CONNECT\_FAILED, XYSSL\_ERR\_NET\_SOCKET\_FAILED, and XYSSL\_ERR\_NET\_UNKNOWN\_HOST.

Referenced by main(), and ssl\_test().

**13.127.1.5 static unsigned short net\_htons (int *port*) [static]**

Definition at line 67 of file net.c.

References buf.

Referenced by net\_bind(), and net\_connect().

**13.127.1.6 static int net\_is\_blocking (void) [static]**

Definition at line 190 of file net.c.

Referenced by net\_accept(), net\_recv(), and net\_send().

**13.127.1.7 int net\_recv (void \* *ctx*, unsigned char \* *buf*, int *len*)**

Read at most 'len' characters. len is updated to reflect the actual number of characters read.

**Returns:**

This function returns the number of bytes received, or a negative error code; XYSSL\_ERR\_NET\_TRY\_AGAIN indicates read() is blocking.

Definition at line 277 of file net.c.

References net\_is\_blocking(), XYSSL\_ERR\_NET\_CONN\_RESET, XYSSL\_ERR\_NET\_RECV\_FAILED, and XYSSL\_ERR\_NET\_TRY\_AGAIN.

Referenced by main(), and ssl\_test().

**13.127.1.8 int net\_send (void \* *ctx*, unsigned char \* *buf*, int *len*)**

Write at most 'len' characters. len is updated to reflect the number of characters *\_not\_* written.

**Returns:**

This function returns the number of bytes sent, or a negative error code; XYSSL\_ERR\_NET\_TRY\_AGAIN indicates write() is blocking.

Definition at line 309 of file net.c.

References net\_is\_blocking(), XYSSL\_ERR\_NET\_CONN\_RESET, XYSSL\_ERR\_NET\_SEND\_FAILED, and XYSSL\_ERR\_NET\_TRY\_AGAIN.

Referenced by main(), and ssl\_test().



#### 13.127.1.9 `int net_set_block (int fd)`

Set the socket blocking.

##### **Returns:**

0 if successful, or a non-zero error code

Definition at line 243 of file net.c.

#### 13.127.1.10 `int net_set_nonblock (int fd)`

Set the socket non-blocking.

##### **Returns:**

0 if successful, or a non-zero error code

Definition at line 253 of file net.c.

Referenced by ssl\_test().

#### 13.127.1.11 `void net_usleep (unsigned long usec)`

Portable usleep helper.

##### **Note:**

Real amount of time slept will not be less than select()'s timeout granularity (typically, 10ms).

Definition at line 266 of file net.c.

### 13.128 `/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/library/padlock.c` File Reference

```
#include "xyssl/config.h"
#include "xyssl/aes.h"
#include "xyssl/padlock.h"
```

## 13.129 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/library/rsa.c File Reference

```
#include "xyssl/config.h"
#include "xyssl/rsa.h"
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
#include "xyssl/sha1.h"
```

### Defines

- #define [KEY\\_LEN](#) 128
- #define [RSA\\_N](#)
- #define [RSA\\_E](#) "10001"
- #define [RSA\\_D](#)
- #define [RSA\\_P](#)
- #define [RSA\\_Q](#)
- #define [RSA\\_DP](#)
- #define [RSA\\_DQ](#)
- #define [RSA\\_QP](#)
- #define [PT\\_LEN](#) 24
- #define [RSA\\_PT](#)

### Functions

- void [rsa\\_init](#) ([rsa\\_context](#) \*ctx, int padding, int hash\_id, int(\*f\_rng)(void \*), void \*p\_rng)  
*Initialize an RSA context.*
- int [rsa\\_gen\\_key](#) ([rsa\\_context](#) \*ctx, int nbits, int exponent)  
*Generate an RSA keypair.*
- int [rsa\\_check\\_pubkey](#) ([rsa\\_context](#) \*ctx)  
*Check a public RSA key.*
- int [rsa\\_check\\_privkey](#) ([rsa\\_context](#) \*ctx)  
*Check a private RSA key.*
- int [rsa\\_public](#) ([rsa\\_context](#) \*ctx, unsigned char \*input, unsigned char \*output)  
*Do an RSA public key operation.*
- int [rsa\\_private](#) ([rsa\\_context](#) \*ctx, unsigned char \*input, unsigned char \*output)  
*Do an RSA private key operation.*
- int [rsa\\_pkcs1\\_encrypt](#) ([rsa\\_context](#) \*ctx, int mode, int ilen, unsigned char \*input, unsigned char \*output)  
*Add the message padding, then do an RSA operation.*

- `int rsa_pkcs1_decrypt (rsa_context *ctx, int mode, int *olen, unsigned char *input, unsigned char *output)`

*Do an RSA operation, then remove the message padding.*

- `int rsa_pkcs1_sign (rsa_context *ctx, int mode, int hash_id, int hashlen, unsigned char *hash, unsigned char *sig)`

*Do a private RSA to sign a message digest.*

- `int rsa_pkcs1_verify (rsa_context *ctx, int mode, int hash_id, int hashlen, unsigned char *hash, unsigned char *sig)`

*Do a public RSA and check the message digest.*

- `void rsa_free (rsa_context *ctx)`

*Free the components of an RSA key.*

- `int rsa_self_test (int verbose)`

*Checkup routine.*

## 13.129.1 Define Documentation

### 13.129.1.1 #define KEY\_LEN 128

Definition at line 573 of file rsa.c.

Referenced by `rsa_self_test()`.

### 13.129.1.2 #define PT\_LEN 24

Definition at line 620 of file rsa.c.

Referenced by `rsa_self_test()`.

### 13.129.1.3 #define RSA\_D

**Value:**

```
"24BF6185468786FDD303083D25E64EFC" \
    "66CA472BC44D253102F8B4A9D3BFA750" \
    "91386C0077937FE33FA3252D28855837" \
    "AE1B484A8A9A45F7EE8C0C634F99E8CD" \
    "DF79C5CE07EE72C7F123142198164234" \
    "CABB724CF78B8173B9F880FC86322407" \
    "AF1FEDFDDE2BEB674CA15F3E81A1521E" \
    "071513A1E85B5DFA031F21ECAE91A34D"
```

Definition at line 586 of file rsa.c.

Referenced by `rsa_self_test()`.

#### 13.129.1.4 #define RSA\_DP

##### Value:

```
"C1ACF567564274FB07A0BBAD5D26E298" \
    "3C94D22288ACD763FD8E5600ED4A702D" \
    "F84198A5F06C2E72236AE490C93F07F8" \
    "3CC559CD27BC2D1CA488811730BB5725"
```

Definition at line 605 of file rsa.c.

Referenced by rsa\_self\_test().

#### 13.129.1.5 #define RSA\_DQ

##### Value:

```
"4959CBF6F8FEF750AEE6977C155579C7" \
    "D8AAEA56749EA28623272E4F7D0592AF" \
    "7C1F1313CAC9471B5C523BFE592F517B" \
    "407A1BD76C164B93DA2D32A383E58357"
```

Definition at line 610 of file rsa.c.

Referenced by rsa\_self\_test().

#### 13.129.1.6 #define RSA\_E "10001"

Definition at line 584 of file rsa.c.

Referenced by rsa\_self\_test().

#### 13.129.1.7 #define RSA\_N

##### Value:

```
"9292758453063D803DD603D5E777D788" \
    "8ED1D5BF35786190FA2F23EBC0848AEA" \
    "DDA92CA6C3D80B32C4D109BE0F36D6AE" \
    "7130B9CED7ACDF54CFC7555AC14EEBAB" \
    "93A89813FBF3C4F8066D2D800F7C38A8" \
    "1AE31942917403FF4946B0A83D3D3E05" \
    "EE57C6F5F5606FB5D4BC6CD34EE0801A" \
    "5E94BB77B07507233A0BC7BAC8F90F79"
```

Definition at line 575 of file rsa.c.

Referenced by rsa\_self\_test().

#### 13.129.1.8 #define RSA\_P

##### Value:

```
"C36D0EB7FCD285223CFB5AABA5BDA3D8" \
    "2C01CAD19EA484A87EA4377637E75500" \
    "FCB2005C5C7DD6EC4AC023CDA285D796" \
    "C3D9E75E1EFC42488BB4F1D13AC30A57"
```

Definition at line 595 of file rsa.c.

Referenced by rsa\_self\_test().

### 13.129.1.9 #define RSA\_PT

**Value:**

```
"\xAA\xBB\xCC\x03\x02\x01\x00\xff\xff\xff\xff" \
"\x11\x22\x33\x0A\x0B\x0C\xCC\xDD\xDD\xDD\xDD"
```

Definition at line 621 of file rsa.c.

Referenced by rsa\_self\_test().

### 13.129.1.10 #define RSA\_Q

**Value:**

```
"C000DF51A7C77AE8D7C7370C1FF55B69" \
"E211C2B9E5DB1ED0BF61D0D9899620F4" \
"910E4168387E3C30AA1E00C339A79508" \
"8452DD96A9A5EA5D9DCA68DA636032AF"
```

Definition at line 600 of file rsa.c.

Referenced by rsa\_self\_test().

### 13.129.1.11 #define RSA\_QP

**Value:**

```
"9AE7FBC99546432DF71896FC239EADAE" \
"F38D18D2B2F0E2DD275AA977E2BF4411" \
"F5A3B2A5D33605AEBBCCBA7FEB9F2D2F" \
"A74206CEC169D74BF5A8C50D6F48EA08"
```

Definition at line 615 of file rsa.c.

Referenced by rsa\_self\_test().

## 13.129.2 Function Documentation

### 13.129.2.1 int rsa\_check\_privkey (rsa\_context \* ctx)

Check a private RSA key.

**Parameters:**

*ctx* RSA context to be checked

**Returns:**

0 if successful, or an XYSSL\_ERR\_RSA\_XXX error code

Definition at line 152 of file rsa.c.

References `rsa_context::D`, `rsa_context::E`, `MPI_CHK`, `mpi_cmp_int()`, `mpi_cmp_mpi()`, `mpi_free()`, `mpi_gcd()`, `mpi_init()`, `mpi_mod_mpi()`, `mpi_mul_mpi()`, `mpi_sub_int()`, `rsa_context::N`, `rsa_context::P`, `rsa_context::Q`, `rsa_check_pubkey()`, and `XYSSL_ERR_RSA_KEY_CHECK_FAILED`.

Referenced by `rsa_decryption()`, `rsa_self_test()`, and `x509parse_key()`.

### 13.129.2.2 `int rsa_check_pubkey (rsa_context * ctx)`

Check a public RSA key.

#### Parameters:

*ctx* RSA context to be checked

#### Returns:

0 if successful, or an `XYSSL_ERR_RSA_XXX` error code

Definition at line 132 of file rsa.c.

References `rsa_context::E`, `mpi_msb()`, `rsa_context::N`, `mpi::p`, and `XYSSL_ERR_RSA_KEY_CHECK_FAILED`.

Referenced by `rsa_check_privkey()`, `rsa_encryption()`, `rsa_self_test()`, and `x509parse_crt()`.

### 13.129.2.3 `void rsa_free (rsa_context * ctx)`

Free the components of an RSA key.

Definition at line 558 of file rsa.c.

References `rsa_context::D`, `rsa_context::DP`, `rsa_context::DQ`, `rsa_context::E`, `mpi_free()`, `rsa_context::N`, `rsa_context::P`, `rsa_context::Q`, `rsa_context::QP`, `rsa_context::RN`, `rsa_context::RP`, and `rsa_context::RQ`.

Referenced by `main()`, `rsa_gen_key()`, `rsa_self_test()`, `ssl_test()`, `x509_free()`, `x509_self_test()`, and `x509parse_key()`.

### 13.129.2.4 `int rsa_gen_key (rsa_context * ctx, int nbits, int exponent)`

Generate an RSA keypair.

#### Parameters:

*ctx* RSA context that will hold the key

*nbits* size of the public key in bits

*exponent* public exponent (e.g., 65537)

#### Note:

[rsa\\_init\(\)](#) must be called beforehand to setup the RSA context (especially `f_rng` and `p_rng`).

#### Returns:

0 if successful, or an `XYSSL_ERR_RSA_XXX` error code

Definition at line 60 of file rsa.c.

References `rsa_context::D`, `rsa_context::DP`, `rsa_context::DQ`, `rsa_context::E`, `rsa_context::f_rng`, `rsa_context::len`, `MPI_CHK`, `mpi_cmp_int()`, `mpi_cmp_mpi()`, `mpi_free()`, `mpi_gcd()`, `mpi_gen_prime()`, `mpi_init()`, `mpi_inv_mod()`, `mpi_lset()`, `mpi_mod_mpi()`, `mpi_msb()`, `mpi_mul_mpi()`, `mpi_sub_int()`, `mpi_swap()`, `rsa_context::N`, `rsa_context::P`, `rsa_context::p_rng`, `rsa_context::Q`, `rsa_context::QP`, `rsa_free()`, `XYSSL_ERR_RSA_BAD_INPUT_DATA`, and `XYSSL_ERR_RSA_KEY_GEN_FAILED`.

Referenced by `generate_RSA_keys_ciphertext()`, `generate_RSA_keys_plaintext()`, and `main()`.

### 13.129.2.5 `void rsa_init (rsa_context * ctx, int padding, int hash_id, int(*) (void *) f_rng, void * p_rng)`

Initialize an RSA context.

#### Parameters:

*ctx* RSA context to be initialized  
*padding* `RSA_PKCS_V15` or `RSA_PKCS_V21`  
*hash\_id* `RSA_PKCS_V21` hash identifier  
*f\_rng* RNG function  
*p\_rng* RNG parameter

#### Note:

The *hash\_id* parameter is actually ignored when using `RSA_PKCS_V15` padding.  
 Currently (xyssl-0.8), `RSA_PKCS_V21` padding is not supported.

Definition at line 40 of file rsa.c.

References `rsa_context::f_rng`, `rsa_context::hash_id`, `rsa_context::p_rng`, and `rsa_context::padding`.

Referenced by `generate_RSA_keys_ciphertext()`, `generate_RSA_keys_plaintext()`, `main()`, `rsa_decryption()`, and `rsa_encryption()`.

### 13.129.2.6 `int rsa_pkcs1_decrypt (rsa_context * ctx, int mode, int * olen, unsigned char * input, unsigned char * output)`

Do an RSA operation, then remove the message padding.

#### Parameters:

*ctx* RSA context  
*mode* `RSA_PUBLIC` or `RSA_PRIVATE`  
*input* buffer holding the encrypted data  
*output* buffer that will hold the plaintext  
*olen* will contain the plaintext length

#### Returns:

0 if successful, or an `XYSSL_ERR_RSA_XXX` error code

#### Note:

The output buffer must be as large as the size of `ctx->N` (eg. 128 bytes if RSA-1024 is used).



Definition at line 326 of file rsa.c.

References `buf`, `int`, `rsa_context::len`, `rsa_context::padding`, `RSA_CRYPT`, `RSA_PKCS_V15`, `rsa_private()`, `rsa_public()`, `RSA_PUBLIC`, `XYSSL_ERR_RSA_BAD_INPUT_DATA`, and `XYSSL_ERR_RSA_INVALID_PADDING`.

Referenced by `rsa_decryption()`, `RSA_private_decrypt()`, `RSA_public_decrypt()`, `rsa_self_test()`, and `ssl_parse_client_key_exchange()`.

### 13.129.2.7 `int rsa_pkcs1_encrypt(rsa_context * ctx, int mode, int ilen, unsigned char * input, unsigned char * output)`

Add the message padding, then do an RSA operation.

#### Parameters:

*ctx* RSA context  
*mode* `RSA_PUBLIC` or `RSA_PRIVATE`  
*ilen* contains the the plaintext length  
*input* buffer holding the data to be encrypted  
*output* buffer that will hold the ciphertext

#### Returns:

0 if successful, or an `XYSSL_ERR_RSA_XXX` error code

#### Note:

The output buffer must be as large as the size of `ctx->N` (eg. 128 bytes if RSA-1024 is used).

Definition at line 280 of file rsa.c.

References `rsa_context::len`, `rsa_context::padding`, `RSA_CRYPT`, `RSA_PKCS_V15`, `rsa_private()`, `rsa_public()`, `RSA_PUBLIC`, `XYSSL_ERR_RSA_BAD_INPUT_DATA`, and `XYSSL_ERR_RSA_INVALID_PADDING`.

Referenced by `rsa_encryption()`, `RSA_private_encrypt()`, `RSA_public_encrypt()`, `rsa_self_test()`, and `ssl_write_client_key_exchange()`.

### 13.129.2.8 `int rsa_pkcs1_sign(rsa_context * ctx, int mode, int hash_id, int hashlen, unsigned char * hash, unsigned char * sig)`

Do a private RSA to sign a message digest.

#### Parameters:

*ctx* RSA context  
*mode* `RSA_PUBLIC` or `RSA_PRIVATE`  
*hash\_id* `RSA_RAW`, `RSA_MD{2,4,5}` or `RSA_SHA{1,256}`  
*hashlen* message digest length (for `RSA_RAW` only)  
*hash* buffer holding the message digest  
*sig* buffer that will hold the ciphertext

**Returns:**

0 if the signing operation was successful, or an XYSSL\_ERR\_RSA\_XXX error code

**Note:**

The "sig" buffer must be as large as the size of ctx->N (eg. 128 bytes if RSA-1024 is used).

Definition at line 379 of file rsa.c.

References ASN1\_HASH\_MDX, ASN1\_HASH\_SHA1, rsa\_context::len, rsa\_context::padding, RSA\_MD2, RSA\_MD4, RSA\_MD5, RSA\_PKCS\_V15, rsa\_private(), rsa\_public(), RSA\_PUBLIC, RSA\_RAW, RSA\_SHA1, RSA\_SIGN, XYSSL\_ERR\_RSA\_BAD\_INPUT\_DATA, and XYSSL\_ERR\_RSA\_INVALID\_PADDING.

Referenced by main(), rsa\_self\_test(), ssl\_write\_certificate\_verify(), and ssl\_write\_server\_key\_exchange().

### 13.129.2.9 **int rsa\_pkcs1\_verify (rsa\_context \* ctx, int mode, int hash\_id, int hashlen, unsigned char \* hash, unsigned char \* sig)**

Do a public RSA and check the message digest.

**Parameters:**

*ctx* points to an RSA public key  
*mode* RSA\_PUBLIC or RSA\_PRIVATE  
*hash\_id* RSA\_RAW, RSA\_MD{2,4,5} or RSA\_SHA{1,256}  
*hashlen* message digest length (for RSA\_RAW only)  
*hash* buffer holding the message digest  
*sig* buffer holding the ciphertext

**Returns:**

0 if the verify operation was successful, or an XYSSL\_ERR\_RSA\_XXX error code

**Note:**

The "sig" buffer must be as large as the size of ctx->N (eg. 128 bytes if RSA-1024 is used).

Definition at line 468 of file rsa.c.

References ASN1\_HASH\_MDX, ASN1\_HASH\_SHA1, buf, int, rsa\_context::len, rsa\_context::padding, RSA\_MD2, RSA\_MD4, RSA\_MD5, RSA\_PKCS\_V15, rsa\_private(), rsa\_public(), RSA\_PUBLIC, RSA\_RAW, RSA\_SHA1, RSA\_SIGN, XYSSL\_ERR\_RSA\_BAD\_INPUT\_DATA, XYSSL\_ERR\_RSA\_INVALID\_PADDING, and XYSSL\_ERR\_RSA\_VERIFY\_FAILED.

Referenced by main(), rsa\_self\_test(), ssl\_parse\_certificate\_verify(), ssl\_parse\_server\_key\_exchange(), and x509parse\_verify().

### 13.129.2.10 **int rsa\_private (rsa\_context \* ctx, unsigned char \* input, unsigned char \* output)**

Do an RSA private key operation.

**Parameters:**

*ctx* RSA context

*input* input buffer  
*output* output buffer

**Returns:**

0 if successful, or an XYSSL\_ERR\_RSA\_XXX error code

**Note:**

The input and output buffers must be large enough (eg. 128 bytes if RSA-1024 is used).

Definition at line 221 of file rsa.c.

References `rsa_context::D`, `rsa_context::DP`, `rsa_context::DQ`, `rsa_context::len`, `mpi_add_mpi()`, `MPI_CHK`, `mpi_cmp_mpi()`, `mpi_exp_mod()`, `mpi_free()`, `mpi_init()`, `mpi_mod_mpi()`, `mpi_mul_mpi()`, `mpi_read_binary()`, `mpi_sub_mpi()`, `mpi_write_binary()`, `rsa_context::N`, `rsa_context::P`, `rsa_context::Q`, `rsa_context::QP`, `rsa_context::RN`, `rsa_context::RP`, `rsa_context::RQ`, `XYSSL_ERR_RSA_BAD_INPUT_DATA`, and `XYSSL_ERR_RSA_PRIVATE_FAILED`.

Referenced by `main()`, `rsa_pkcs1_decrypt()`, `rsa_pkcs1_encrypt()`, `rsa_pkcs1_sign()`, and `rsa_pkcs1_verify()`.

**13.129.2.11 int rsa\_public (rsa\_context \* ctx, unsigned char \* input, unsigned char \* output)**

Do an RSA public key operation.

**Parameters:**

*ctx* RSA context  
*input* input buffer  
*output* output buffer

**Returns:**

0 if successful, or an XYSSL\_ERR\_RSA\_XXX error code

**Note:**

This function does NOT take care of message padding. Also, be sure to set `input[0] = 0`.  
The input and output buffers must be large enough (eg. 128 bytes if RSA-1024 is used).

Definition at line 187 of file rsa.c.

References `rsa_context::E`, `rsa_context::len`, `MPI_CHK`, `mpi_cmp_mpi()`, `mpi_exp_mod()`, `mpi_free()`, `mpi_init()`, `mpi_read_binary()`, `mpi_write_binary()`, `rsa_context::N`, `rsa_context::RN`, `XYSSL_ERR_RSA_BAD_INPUT_DATA`, and `XYSSL_ERR_RSA_PUBLIC_FAILED`.

Referenced by `main()`, `rsa_pkcs1_decrypt()`, `rsa_pkcs1_encrypt()`, `rsa_pkcs1_sign()`, and `rsa_pkcs1_verify()`.

**13.129.2.12 int rsa\_self\_test (int verbose)**

Checkup routine.

**Returns:**

0 if successful, or 1 if the test failed

Definition at line 627 of file rsa.c.

References `rsa_context::D`, `rsa_context::DP`, `rsa_context::DQ`, `rsa_context::E`, `KEY_LEN`, `rsa_context::len`, `mpi_read_string()`, `rsa_context::N`, `rsa_context::P`, `PT_LEN`, `rsa_context::Q`, `rsa_context::QP`, `rsa_check_privkey()`, `rsa_check_pubkey()`, `RSA_D`, `RSA_DP`, `RSA_DQ`, `RSA_E`, `rsa_free()`, `RSA_N`, `RSA_P`, `rsa_pkcs1_decrypt()`, `rsa_pkcs1_encrypt()`, `rsa_pkcs1_sign()`, `rsa_pkcs1_verify()`, `RSA_PRIVATE`, `RSA_PT`, `RSA_PUBLIC`, `RSA_Q`, `RSA_QP`, `RSA_SHA1`, and `sha1()`.

Referenced by `main()`.

## 13.130 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/library/sha1.c File Reference

```
#include "xyssl/config.h"
#include "xyssl/sha1.h"
#include <string.h>
#include <stdio.h>
```

### Defines

- #define **S**(x, n) ((x << n) | ((x & 0xFFFFFFFF) >> (32 - n)))
- #define **R**(t)
- #define **P**(a, b, c, d, e, x)
- #define **F**(x, y, z) (z ^ (x & (y ^ z)))
- #define **K** 0x5A827999
- #define **F**(x, y, z) (x ^ y ^ z)
- #define **K** 0x6ED9EBA1
- #define **F**(x, y, z) ((x & y) | (z & (x | y)))
- #define **K** 0x8F1BBCDC
- #define **F**(x, y, z) (x ^ y ^ z)
- #define **K** 0xCA62C1D6

### Functions

- void **sha1\_starts** (**sha1\_context** \*ctx)  
*SHA-1 context setup.*
- static void **sha1\_process** (**sha1\_context** \*ctx, unsigned char data[64])
- void **sha1\_update** (**sha1\_context** \*ctx, unsigned char \*input, **int** ilen)  
*SHA-1 process buffer.*
- void **sha1\_finish** (**sha1\_context** \*ctx, unsigned char output[20])  
*SHA-1 final digest.*
- void **sha1** (unsigned char \*input, **int** ilen, unsigned char output[20])  
*Output = SHA-1( input buffer ).*
- **int** **sha1\_file** (char \*path, unsigned char output[20])  
*Output = SHA-1( file contents ).*
- void **sha1\_hmac\_starts** (**sha1\_context** \*ctx, unsigned char \*key, **int** keylen)  
*SHA-1 HMAC context setup.*
- void **sha1\_hmac\_update** (**sha1\_context** \*ctx, unsigned char \*input, **int** ilen)  
*SHA-1 HMAC process buffer.*
- void **sha1\_hmac\_finish** (**sha1\_context** \*ctx, unsigned char output[20])

*SHA-1 HMAC final digest.*

- void `sha1_hmac` (unsigned char \*key, int keylen, unsigned char \*input, int ilen, unsigned char output[20])

*Output = HMAC-SHA-1( hmac key, input buffer ).*

- int `sha1_self_test` (int verbose)

*Checkup routine.*

## Variables

- static const unsigned char `sha1_padding` [64]
- static unsigned char `sha1_test_buf` [3][57]
- static const int `sha1_test_buflen` [3]
- static const unsigned char `sha1_test_sum` [3][20]
- static unsigned char `sha1_hmac_test_key` [7][26]
- static const int `sha1_hmac_test_keylen` [7]
- static unsigned char `sha1_hmac_test_buf` [7][74]
- static const int `sha1_hmac_test_buflen` [7]
- static const unsigned char `sha1_hmac_test_sum` [7][20]

### 13.130.1 Define Documentation

**13.130.1.1** `#define F(x, y, z) (x ^ y ^ z)`

**13.130.1.2** `#define F(x, y, z) ((x & y) | (z & (x | y)))`

**13.130.1.3** `#define F(x, y, z) (x ^ y ^ z)`

**13.130.1.4** `#define F(x, y, z) (z ^ (x & (y ^ z)))`

**13.130.1.5** `#define K 0xCA62C1D6`

**13.130.1.6** `#define K 0x8F1BBCDC`

**13.130.1.7** `#define K 0x6ED9EBA1`

**13.130.1.8** `#define K 0x5A827999`

Referenced by `sha4_process()`.

**13.130.1.9** `#define P(a, b, c, d, e, x)`

**Value:**

```
{
    e += S(a, 5) + F(b, c, d) + K + x; b = S(b, 30);
}
```

### 13.130.1.10 #define R(t)

**Value:**

```
(
    temp = W[(t - 3) & 0x0F] ^ W[(t - 8) & 0x0F] ^ \
           W[(t - 14) & 0x0F] ^ W[t & 0x0F], \
    ( W[t & 0x0F] = S(temp, 1) ) \
)
```

Referenced by mpi\_is\_prime(), sha1\_process(), and sha2\_process().

### 13.130.1.11 #define S(x, n) ((x << n) | ((x & 0xFFFFFFFF) >> (32 - n)))

## 13.130.2 Function Documentation

### 13.130.2.1 void sha1 (unsigned char \* *input*, int *ilen*, unsigned char *output*[20])

Output = SHA-1( *input* buffer ).

**Parameters:**

- input* buffer holding the data
- ilen* length of the input data
- output* SHA-1 checksum result

Definition at line 313 of file sha1.c.

References sha1\_finish(), sha1\_starts(), and sha1\_update().

Referenced by main(), rsa\_self\_test(), sha1\_hmac\_starts(), ssl\_calc\_verify(), ssl\_derive\_keys(), ssl\_mac\_sha1(), ssl\_parse\_finished(), ssl\_parse\_server\_key\_exchange(), ssl\_write\_finished(), ssl\_write\_server\_key\_exchange(), and x509\_hash().

### 13.130.2.2 int sha1\_file (char \* *path*, unsigned char *output*[20])

Output = SHA-1( *file* contents ).

**Parameters:**

- path* input file name
- output* SHA-1 checksum result

**Returns:**

- 0 if successful, 1 if fopen failed, or 2 if fread failed

Definition at line 327 of file sha1.c.

References buf, f, sha1\_finish(), sha1\_starts(), and sha1\_update().

Referenced by main(), and sha1\_wrapper().

**13.130.2.3 void sha1\_finish (sha1\_context \* ctx, unsigned char output[20])**

SHA-1 final digest.

**Parameters:**

*ctx* SHA-1 context

*output* SHA-1 checksum result

Definition at line 284 of file sha1.c.

References PUT\_ULONG\_BE, sha1\_padding, sha1\_update(), sha1\_context::state, and sha1\_context::total.

Referenced by sha1(), sha1\_file(), sha1\_hmac\_finish(), sha1\_self\_test(), ssl\_calc\_finished(), ssl\_calc\_verify(), ssl\_derive\_keys(), ssl\_mac\_sha1(), ssl\_parse\_server\_key\_exchange(), and ssl\_write\_server\_key\_exchange().

**13.130.2.4 void sha1\_hmac (unsigned char \* key, int keylen, unsigned char \* input, int ilen, unsigned char output[20])**

Output = HMAC-SHA-1( hmac key, input buffer ).

**Parameters:**

*key* HMAC secret key

*keylen* length of the HMAC key

*input* buffer holding the data

*ilen* length of the input data

*output* HMAC-SHA-1 result

Definition at line 413 of file sha1.c.

References sha1\_hmac\_finish(), sha1\_hmac\_starts(), and sha1\_hmac\_update().

Referenced by ssl\_decrypt\_buf(), ssl\_encrypt\_buf(), and tls1\_prf().

**13.130.2.5 void sha1\_hmac\_finish (sha1\_context \* ctx, unsigned char output[20])**

SHA-1 HMAC final digest.

**Parameters:**

*ctx* HMAC context

*output* SHA-1 HMAC checksum result

Definition at line 397 of file sha1.c.

References sha1\_context::opad, sha1\_finish(), sha1\_starts(), and sha1\_update().

Referenced by sha1\_hmac(), and sha1\_self\_test().

**13.130.2.6 void sha1\_hmac\_starts (sha1\_context \* ctx, unsigned char \* key, int keylen)**

SHA-1 HMAC context setup.



**Parameters:**

*ctx* HMAC context to be initialized  
*key* HMAC secret key  
*keylen* length of the HMAC key

Definition at line 359 of file sha1.c.

References sha1\_context::ipad, sha1\_context::opad, sha1(), sha1\_starts(), and sha1\_update().

Referenced by sha1\_hmac(), and sha1\_self\_test().

**13.130.2.7 void sha1\_hmac\_update (sha1\_context \* ctx, unsigned char \* input, int ilen)**

SHA-1 HMAC process buffer.

**Parameters:**

*ctx* HMAC context  
*input* buffer holding the data  
*ilen* length of the input data

Definition at line 389 of file sha1.c.

References sha1\_update().

Referenced by sha1\_hmac(), and sha1\_self\_test().

**13.130.2.8 static void sha1\_process (sha1\_context \* ctx, unsigned char data[64]) [static]**

Definition at line 73 of file sha1.c.

References GET\_ULONG\_BE, P, R, and sha1\_context::state.

Referenced by sha1\_update().

**13.130.2.9 int sha1\_self\_test (int verbose)**

Checkup routine.

**Returns:**

0 if successful, or 1 if the test failed

Definition at line 521 of file sha1.c.

References buf, sha1\_finish(), sha1\_hmac\_finish(), sha1\_hmac\_starts(), sha1\_hmac\_test\_buf, sha1\_hmac\_test\_buflen, sha1\_hmac\_test\_key, sha1\_hmac\_test\_keylen, sha1\_hmac\_test\_sum, sha1\_hmac\_update(), sha1\_starts(), sha1\_test\_buf, sha1\_test\_buflen, sha1\_test\_sum, and sha1\_update().

Referenced by main().

**13.130.2.10 void sha1\_starts (sha1\_context \* ctx)**

SHA-1 context setup.

**Parameters:**

*ctx* context to be initialized

Definition at line 61 of file sha1.c.

References sha1\_context::state, and sha1\_context::total.

Referenced by sha1(), sha1\_file(), sha1\_hmac\_finish(), sha1\_hmac\_starts(), sha1\_self\_test(), ssl\_calc\_finished(), ssl\_calc\_verify(), ssl\_derive\_keys(), ssl\_init(), ssl\_mac\_sha1(), ssl\_parse\_server\_key\_exchange(), and ssl\_write\_server\_key\_exchange().

**13.130.2.11 void sha1\_update (sha1\_context \* ctx, unsigned char \* input, int ilen)**

SHA-1 process buffer.

**Parameters:**

*ctx* SHA-1 context

*input* buffer holding the data

*ilen* length of the input data

Definition at line 232 of file sha1.c.

References sha1\_context::buffer, sha1\_process(), and sha1\_context::total.

Referenced by sha1(), sha1\_file(), sha1\_finish(), sha1\_hmac\_finish(), sha1\_hmac\_starts(), sha1\_hmac\_update(), sha1\_self\_test(), ssl\_calc\_finished(), ssl\_calc\_verify(), ssl\_derive\_keys(), ssl\_mac\_sha1(), ssl\_parse\_client\_hello(), ssl\_parse\_server\_key\_exchange(), ssl\_read\_record(), ssl\_write\_record(), and ssl\_write\_server\_key\_exchange().

**13.130.3 Variable Documentation****13.130.3.1 unsigned char sha1\_hmac\_test\_buf[7][74] [static]****Initial value:**

```
{
    { "Hi There" },
    { "what do ya want for nothing?" },
    { "\xDD\xDD\xDD\xDD\xDD\xDD\xDD\xDD\xDD\xDD"
      "\xDD\xDD\xDD\xDD\xDD\xDD\xDD\xDD\xDD\xDD"
      "\xDD\xDD\xDD\xDD\xDD\xDD\xDD\xDD\xDD\xDD"
      "\xDD\xDD\xDD\xDD\xDD\xDD\xDD\xDD\xDD\xDD"
      "\xDD\xDD\xDD\xDD\xDD\xDD\xDD\xDD\xDD\xDD" },
    { "\xCD\xCD\xCD\xCD\xCD\xCD\xCD\xCD\xCD\xCD"
      "\xCD\xCD\xCD\xCD\xCD\xCD\xCD\xCD\xCD\xCD"
      "\xCD\xCD\xCD\xCD\xCD\xCD\xCD\xCD\xCD\xCD"
      "\xCD\xCD\xCD\xCD\xCD\xCD\xCD\xCD\xCD\xCD"
      "\xCD\xCD\xCD\xCD\xCD\xCD\xCD\xCD\xCD\xCD" },
    { "Test With Truncation" },
    { "Test Using Larger Than Block-Size Key - Hash Key First" },
    { "Test Using Larger Than Block-Size Key and Larger"
      " Than One Block-Size Data" }
}
```

Definition at line 475 of file sha1.c.

Referenced by sha1\_self\_test().

**Initial value:**

Definition at line 495 of file sha1.c.

**Initial value:**

Definition at line 455 of file sha1.c.

**Initial value:**

Definition at line 470 of file sha1.c.

**Initial value:**

Generated on Wed Mar 17 15:31:55 2010 for Mobile-C by Doxygen

```

    { 0x12, 0x5D, 0x73, 0x42, 0xB9, 0xAC, 0x11, 0xCD, 0x91, 0xA3,
      0x9A, 0xF4, 0x8A, 0xA1, 0x7B, 0x4F, 0x63, 0xF1, 0x75, 0xD3 },
    { 0x4C, 0x90, 0x07, 0xF4, 0x02, 0x62, 0x50, 0xC6, 0xBC, 0x84,
      0x14, 0xF9, 0xBF, 0x50, 0xC8, 0x6C, 0x2D, 0x72, 0x35, 0xDA },
    { 0x4C, 0x1A, 0x03, 0x42, 0x4B, 0x55, 0xE0, 0x7F, 0xE7, 0xF2,
      0x7B, 0xE1 },
    { 0xAA, 0x4A, 0xE5, 0xE1, 0x52, 0x72, 0xD0, 0x0E, 0x95, 0x70,
      0x56, 0x37, 0xCE, 0x8A, 0x3B, 0x55, 0xED, 0x40, 0x21, 0x12 },
    { 0xE8, 0xE9, 0x9D, 0x0F, 0x45, 0x23, 0x7D, 0x78, 0x6D, 0x6B,
      0xBA, 0xA7, 0x96, 0x5C, 0x78, 0x08, 0xBB, 0xFF, 0x1A, 0x91 }
}

```

Definition at line 500 of file sha1.c.

Referenced by sha1\_self\_test().

### 13.130.3.6 const unsigned char sha1\_padding[64] [static]

**Initial value:**

```

{
    0x80, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
}

```

Definition at line 273 of file sha1.c.

Referenced by sha1\_finish().

### 13.130.3.7 unsigned char sha1\_test\_buf[3][57] [static]

**Initial value:**

```

{
    { "abc" },
    { "abcdefghijklmnpq" },
    { "" }
}

```

Definition at line 430 of file sha1.c.

Referenced by sha1\_self\_test().

### 13.130.3.8 const int sha1\_test\_buflen[3] [static]

**Initial value:**

```

{
    3, 56, 1000
}

```

Definition at line 437 of file sha1.c.

Referenced by sha1\_self\_test().

### 13.130.3.9 `const unsigned char sha1_test_sum[3][20] [static]`

**Initial value:**

```
{
    { 0xA9, 0x99, 0x3E, 0x36, 0x47, 0x06, 0x81, 0x6A, 0xBA, 0x3E,
      0x25, 0x71, 0x78, 0x50, 0xC2, 0x6C, 0x9C, 0xD0, 0xD8, 0x9D },
    { 0x84, 0x98, 0x3E, 0x44, 0x1C, 0x3B, 0xD2, 0x6E, 0xBA, 0xAE,
      0x4A, 0xA1, 0xF9, 0x51, 0x29, 0xE5, 0xE5, 0x46, 0x70, 0xF1 },
    { 0x34, 0xAA, 0x97, 0x3C, 0xD4, 0xC4, 0xDA, 0xA4, 0xF6, 0x1E,
      0xEB, 0x2B, 0xDB, 0xAD, 0x27, 0x31, 0x65, 0x34, 0x01, 0x6F }
}
```

Definition at line 442 of file sha1.c.

Referenced by sha1\_self\_test().

## 13.131 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/library/sha2.c File Reference

```
#include "xyssl/config.h"
#include "xyssl/sha2.h"
#include <string.h>
#include <stdio.h>
```

### Defines

- #define [SHR](#)(x, n) ((x & 0xFFFFFFFF) >> n)
- #define [ROTR](#)(x, n) (SHR(x,n) | (x << (32 - n)))
- #define [S0](#)(x) (ROTR(x, 7) ^ ROTR(x,18) ^ SHR(x, 3))
- #define [S1](#)(x) (ROTR(x,17) ^ ROTR(x,19) ^ SHR(x,10))
- #define [S2](#)(x) (ROTR(x, 2) ^ ROTR(x,13) ^ ROTR(x,22))
- #define [S3](#)(x) (ROTR(x, 6) ^ ROTR(x,11) ^ ROTR(x,25))
- #define [F0](#)(x, y, z) ((x & y) | (z & (x | y)))
- #define [F1](#)(x, y, z) (z ^ (x & (y ^ z)))
- #define [R](#)(t)
- #define [P](#)(a, b, c, d, e, [f](#), g, h, x, [K](#))

### Functions

- void [sha2\\_starts](#) ([sha2\\_context](#) \*ctx, [int](#) is224)  
*SHA-256 context setup.*
- static void [sha2\\_process](#) ([sha2\\_context](#) \*ctx, unsigned char data[64])
- void [sha2\\_update](#) ([sha2\\_context](#) \*ctx, unsigned char \*input, [int](#) ilen)  
*SHA-256 process buffer.*
- void [sha2\\_finish](#) ([sha2\\_context](#) \*ctx, unsigned char output[32])  
*SHA-256 final digest.*
- void [sha2](#) (unsigned char \*input, [int](#) ilen, unsigned char output[32], [int](#) is224)  
*Output = SHA-256( input buffer ).*
- [int](#) [sha2\\_file](#) (char \*path, unsigned char output[32], [int](#) is224)  
*Output = SHA-256( file contents ).*
- void [sha2\\_hmac\\_starts](#) ([sha2\\_context](#) \*ctx, unsigned char \*key, [int](#) keylen, [int](#) is224)  
*SHA-256 HMAC context setup.*
- void [sha2\\_hmac\\_update](#) ([sha2\\_context](#) \*ctx, unsigned char \*input, [int](#) ilen)  
*SHA-256 HMAC process buffer.*
- void [sha2\\_hmac\\_finish](#) ([sha2\\_context](#) \*ctx, unsigned char output[32])  
*SHA-256 HMAC final digest.*

- void [sha2\\_hmac](#) (unsigned char \*key, int keylen, unsigned char \*input, int ilen, unsigned char output[32], int is224)

*Output = HMAC-SHA-256( hmac key, input buffer ).*

- int [sha2\\_self\\_test](#) (int verbose)

*Checkup routine.*

## Variables

- static const unsigned char [sha2\\_padding](#) [64]
- static unsigned char [sha2\\_test\\_buf](#) [3][57]
- static const int [sha2\\_test\\_buflen](#) [3]
- static const unsigned char [sha2\\_test\\_sum](#) [6][32]
- static unsigned char [sha2\\_hmac\\_test\\_key](#) [7][26]
- static const int [sha2\\_hmac\\_test\\_keylen](#) [7]
- static unsigned char [sha2\\_hmac\\_test\\_buf](#) [7][153]
- static const int [sha2\\_hmac\\_test\\_buflen](#) [7]
- static const unsigned char [sha2\\_hmac\\_test\\_sum](#) [14][32]

### 13.131.1 Define Documentation

**13.131.1.1** `#define F0(x, y, z) ((x & y) | (z & (x | y)))`

**13.131.1.2** `#define F1(x, y, z) (z ^ (x & (y ^ z)))`

**13.131.1.3** `#define P(a, b, c, d, e, f, g, h, x, K)`

**Value:**

```
{
    temp1 = h + S3(e) + F1(e, f, g) + K + x;      \
    temp2 = S2(a) + F0(a, b, c);                  \
    d += temp1; h = temp1 + temp2;                 \
}
```

**13.131.1.4** `#define R(t)`

**Value:**

```
(
    W[t] = S1(W[t - 2]) + W[t - 7] +             \
    S0(W[t - 15]) + W[t - 16]                     \
)
```

**13.131.1.5** `#define ROTR(x, n) (SHR(x,n) | (x << (32 - n)))`

**13.131.1.6** `#define S0(x) (ROTR(x, 7) ^ ROTR(x,18) ^ SHR(x, 3))`

Referenced by [sha4\\_process\(\)](#).

**13.131.1.7** `#define S1(x) (ROTR(x,17) ^ ROTR(x,19) ^ SHR(x,10))`

Referenced by sha4\_process(), and tls1\_prf().

**13.131.1.8** `#define S2(x) (ROTR(x, 2) ^ ROTR(x,13) ^ ROTR(x,22))`

Referenced by tls1\_prf().

**13.131.1.9** `#define S3(x) (ROTR(x, 6) ^ ROTR(x,11) ^ ROTR(x,25))`

**13.131.1.10** `#define SHR(x, n) ((x & 0xFFFFFFFF) >> n)`

## 13.131.2 Function Documentation

**13.131.2.1** `void sha2 (unsigned char *input, int ilen, unsigned char output[32], int is224)`

Output = SHA-256( input buffer ).

### Parameters:

*input* buffer holding the data

*ilen* length of the input data

*output* SHA-224/256 checksum result

*is224* 0 = use SHA256, 1 = use SHA224

Definition at line 314 of file sha2.c.

References sha2\_finish(), sha2\_starts(), and sha2\_update().

Referenced by main(), and sha2\_hmac\_starts().

**13.131.2.2** `int sha2_file (char *path, unsigned char output[32], int is224)`

Output = SHA-256( file contents ).

### Parameters:

*path* input file name

*output* SHA-224/256 checksum result

*is224* 0 = use SHA256, 1 = use SHA224

### Returns:

0 if successful, 1 if fopen failed, or 2 if fread failed

Definition at line 329 of file sha2.c.

References buf, f, sha2\_finish(), sha2\_starts(), and sha2\_update().

Referenced by sha2\_wrapper().



### 13.131.2.3 void sha2\_finish (sha2\_context \* ctx, unsigned char output[32])

SHA-256 final digest.

#### Parameters:

*ctx* SHA-256 context

*output* SHA-224/256 checksum result

Definition at line 280 of file sha2.c.

References sha2\_context::is224, PUT\_ULONG\_BE, sha2\_padding, sha2\_update(), sha2\_context::state, and sha2\_context::total.

Referenced by aes\_en\_de(), main(), sha2(), sha2\_file(), sha2\_hmac\_finish(), and sha2\_self\_test().

### 13.131.2.4 void sha2\_hmac (unsigned char \* key, int keylen, unsigned char \* input, int ilen, unsigned char output[32], int is224)

Output = HMAC-SHA-256( hmac key, input buffer ).

#### Parameters:

*key* HMAC secret key

*keylen* length of the HMAC key

*input* buffer holding the data

*ilen* length of the input data

*output* HMAC-SHA-224/256 result

*is224* 0 = use SHA256, 1 = use SHA224

Definition at line 420 of file sha2.c.

References sha2\_hmac\_finish(), sha2\_hmac\_starts(), and sha2\_hmac\_update().

### 13.131.2.5 void sha2\_hmac\_finish (sha2\_context \* ctx, unsigned char output[32])

SHA-256 HMAC final digest.

#### Parameters:

*ctx* HMAC context

*output* SHA-224/256 HMAC checksum result

Definition at line 400 of file sha2.c.

References sha2\_context::is224, sha2\_context::opad, sha2\_finish(), sha2\_starts(), and sha2\_update().

Referenced by aes\_en\_de(), main(), sha2\_hmac(), and sha2\_self\_test().

### 13.131.2.6 void sha2\_hmac\_starts (sha2\_context \* ctx, unsigned char \* key, int keylen, int is224)

SHA-256 HMAC context setup.

**Parameters:**

*ctx* HMAC context to be initialized  
*key* HMAC secret key  
*keylen* length of the HMAC key  
*is224* 0 = use SHA256, 1 = use SHA224

Definition at line 361 of file sha2.c.

References sha2\_context::ipad, sha2\_context::opad, sha2(), sha2\_starts(), and sha2\_update().

Referenced by aes\_en\_de(), main(), sha2\_hmac(), and sha2\_self\_test().

**13.131.2.7 void sha2\_hmac\_update (sha2\_context \* ctx, unsigned char \* input, int ilen)**

SHA-256 HMAC process buffer.

**Parameters:**

*ctx* HMAC context  
*input* buffer holding the data  
*ilen* length of the input data

Definition at line 392 of file sha2.c.

References sha2\_update().

Referenced by aes\_en\_de(), main(), sha2\_hmac(), and sha2\_self\_test().

**13.131.2.8 static void sha2\_process (sha2\_context \* ctx, unsigned char data[64]) [static]**

Definition at line 94 of file sha2.c.

References F, GET\_ULONG\_BE, P, R, and sha2\_context::state.

Referenced by sha2\_update().

**13.131.2.9 int sha2\_self\_test (int verbose)**

Checkup routine.

**Returns:**

0 if successful, or 1 if the test failed

Definition at line 599 of file sha2.c.

References buf, sha2\_finish(), sha2\_hmac\_finish(), sha2\_hmac\_starts(), sha2\_hmac\_test\_buf, sha2\_hmac\_test\_buflen, sha2\_hmac\_test\_key, sha2\_hmac\_test\_keylen, sha2\_hmac\_test\_sum, sha2\_hmac\_update(), sha2\_starts(), sha2\_test\_buf, sha2\_test\_buflen, sha2\_test\_sum, and sha2\_update().

Referenced by main().

## SHA-256 context setup.

**ctx** context to be initialized

**is224** 0 = use SHA256, 1 = use SHA224

References sha2\_context::is224, sha2\_context::state, and sha2\_context::total.

Referenced by `aes_en_de()`, `main()`, `sha2()`, `sha2_file()`, `sha2_hmac_finish()`, `sha2_hmac_starts()`, and `sha2_self_test()`.

## SHA-256 process buffer.

*ctx* SHA-256 context  
*input* buffer holding the data  
*ilen* length of the input data

References sha2\_context::buffer, sha2\_process(), and sha2\_context::total.

Referenced by `aes_en_de()`, `main()`, `sha2()`, `sha2_file()`, `sha2_finish()`, `sha2_hmac_finish()`, `sha2_hmac_starts()`, `sha2_hmac_update()`, and `sha2_self_test()`.

**13.131.3.1 unsigned char sha2\_hmac\_test\_buf[7][153] [static]**

```
{
  { "Hi There" },
  { "what do ya want for nothing?" },
  { "\xDD\xDD\xDD\xDD\xDD\xDD\xDD\xDD\xDD\xDD"
    "\xDD\xDD\xDD\xDD\xDD\xDD\xDD\xDD\xDD\xDD"
    "\xDD\xDD\xDD\xDD\xDD\xDD\xDD\xDD\xDD\xDD"
    "\xDD\xDD\xDD\xDD\xDD\xDD\xDD\xDD\xDD\xDD"
    "\xDD\xDD\xDD\xDD\xDD\xDD\xDD\xDD\xDD\xDD" },
  { "\xCD\xCD\xCD\xCD\xCD\xCD\xCD\xCD\xCD\xCD"
    "\xCD\xCD\xCD\xCD\xCD\xCD\xCD\xCD\xCD\xCD"
    "\xCD\xCD\xCD\xCD\xCD\xCD\xCD\xCD\xCD\xCD"
    "\xCD\xCD\xCD\xCD\xCD\xCD\xCD\xCD\xCD\xCD"
    "\xCD\xCD\xCD\xCD\xCD\xCD\xCD\xCD\xCD\xCD" },
  { "Test With Truncation" },
  { "Test Using Larger Than Block-Size Key - Hash Key First" },
  { "This is a test using a larger than block-size key "
    "and a larger than block-size data. The key needs to "
    "be hashed before being used by the HMAC algorithm." }
}
```

Definition at line 507 of file sha2.c.

Referenced by sha2\_self\_test().

### 13.131.3.2 `const int sha2_hmac_test_buflen[7] [static]`

**Initial value:**

```
{
    8, 28, 50, 50, 20, 54, 152
}
```

Definition at line 528 of file sha2.c.

Referenced by sha2\_self\_test().

### 13.131.3.3 `unsigned char sha2_hmac_test_key[7][26] [static]`

**Initial value:**

```
{
    { "\x0B\x0B\x0B\x0B\x0B\x0B\x0B\x0B\x0B\x0B\x0B\x0B\x0B\x0B\x0B\x0B"
      "\x0B\x0B\x0B\x0B" },
    { "Jefe" },
    { "\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA"
      "\xAA\xAA\xAA" },
    { "\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0A\x0B\x0C\x0D\x0E\x0F\x10"
      "\x11\x12\x13\x14\x15\x16\x17\x18\x19" },
    { "\x0C\x0C\x0C\x0C\x0C\x0C\x0C\x0C\x0C\x0C\x0C\x0C\x0C\x0C\x0C"
      "\x0C\x0C\x0C\x0C" },
    { "" },
    { "" }
}
```

Definition at line 487 of file sha2.c.

Referenced by sha2\_self\_test().

### 13.131.3.4 `const int sha2_hmac_test_keylen[7] [static]`

**Initial value:**

```
{
    20, 4, 20, 25, 20, 131, 131
}
```

Definition at line 502 of file sha2.c.

Referenced by sha2\_self\_test().

### 13.131.3.5 `const unsigned char sha2_hmac_test_sum[14][32] [static]`

Definition at line 533 of file sha2.c.

Referenced by sha2\_self\_test().

### 13.131.3.6 `const unsigned char sha2_padding[64] [static]`

**Initial value:**

```
{
    0x80, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
}
```

Definition at line 269 of file sha2.c.

Referenced by sha2\_finish().

### 13.131.3.7 `unsigned char sha2_test_buf[3][57] [static]`

**Initial value:**

```
{
    { "abc" },
    { "abcdbcdecdefdefgefghfghighijhijkijkljklmklmnlmnomnopnopq" },
    { "" }
}
```

Definition at line 437 of file sha2.c.

Referenced by sha2\_self\_test().

### 13.131.3.8 `const int sha2_test_bufllen[3] [static]`

**Initial value:**

```
{
    3, 56, 1000
}
```

Definition at line 444 of file sha2.c.

Referenced by sha2\_self\_test().

### 13.131.3.9 `const unsigned char sha2_test_sum[6][32] [static]`

**Initial value:**

```
{
    { 0x23, 0x09, 0x7D, 0x22, 0x34, 0x05, 0xD8, 0x22,
      0x86, 0x42, 0xA4, 0x77, 0xBD, 0xA2, 0x55, 0xB3,
      0x2A, 0xAD, 0xBC, 0xE4, 0xBD, 0xA0, 0xB3, 0xF7,
      0xE3, 0x6C, 0x9D, 0xA7 },
    { 0x75, 0x38, 0x8B, 0x16, 0x51, 0x27, 0x76, 0xCC,
      0x5D, 0xBA, 0x5D, 0xA1, 0xFD, 0x89, 0x01, 0x50,
      0xB0, 0xC6, 0x45, 0x5C, 0xB4, 0xF5, 0x8B, 0x19,
      0x52, 0x52, 0x25, 0x25 }
}
```

```
{ 0x20, 0x79, 0x46, 0x55, 0x98, 0x0C, 0x91, 0xD8,
  0xBB, 0xB4, 0xC1, 0xEA, 0x97, 0x61, 0x8A, 0x4B,
  0xF0, 0x3F, 0x42, 0x58, 0x19, 0x48, 0xB2, 0xEE,
  0x4E, 0xE7, 0xAD, 0x67 },

{ 0xBA, 0x78, 0x16, 0xBF, 0x8F, 0x01, 0xCF, 0xEA,
  0x41, 0x41, 0x40, 0xDE, 0x5D, 0xAE, 0x22, 0x23,
  0xB0, 0x03, 0x61, 0xA3, 0x96, 0x17, 0x7A, 0x9C,
  0xB4, 0x10, 0xFF, 0x61, 0xF2, 0x00, 0x15, 0xAD },
{ 0x24, 0x8D, 0x6A, 0x61, 0xD2, 0x06, 0x38, 0xB8,
  0xE5, 0xC0, 0x26, 0x93, 0x0C, 0x3E, 0x60, 0x39,
  0xA3, 0x3C, 0xE4, 0x59, 0x64, 0xFF, 0x21, 0x67,
  0xF6, 0xEC, 0xED, 0xD4, 0x19, 0xDB, 0x06, 0xC1 },
{ 0xCD, 0xC7, 0x6E, 0x5C, 0x99, 0x14, 0xFB, 0x92,
  0x81, 0xA1, 0xC7, 0xE2, 0x84, 0xD7, 0x3E, 0x67,
  0xF1, 0x80, 0x9A, 0x48, 0xA4, 0x97, 0x20, 0x0E,
  0x04, 0x6D, 0x39, 0xCC, 0xC7, 0x11, 0x2C, 0xD0 }
}
```

Definition at line 449 of file sha2.c.

Referenced by sha2\_self\_test().

## 13.132 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/library/sha4.c File Reference

```
#include "xyssl/config.h"
#include "xyssl/sha4.h"
#include <string.h>
#include <stdio.h>
```

### Defines

- #define [GET\\_UINT64\\_BE](#)(n, b, i)
- #define [PUT\\_UINT64\\_BE](#)(n, b, i)
- #define [SHR](#)(x, n) (x >> n)
- #define [ROTR](#)(x, n) (SHR(x, n) | (x << (64 - n)))
- #define [S0](#)(x) (ROTR(x, 1) ^ ROTR(x, 8) ^ SHR(x, 7))
- #define [S1](#)(x) (ROTR(x, 19) ^ ROTR(x, 61) ^ SHR(x, 6))
- #define [S2](#)(x) (ROTR(x, 28) ^ ROTR(x, 34) ^ ROTR(x, 39))
- #define [S3](#)(x) (ROTR(x, 14) ^ ROTR(x, 18) ^ ROTR(x, 41))
- #define [F0](#)(x, y, z) ((x & y) | (z & (x | y)))
- #define [F1](#)(x, y, z) (z ^ (x & (y ^ z)))
- #define [P](#)(a, b, c, d, e, f, g, h, x, K)

### Functions

- void [sha4\\_starts](#) ([sha4\\_context](#) \*ctx, int is384)  
*SHA-512 context setup.*
- static void [sha4\\_process](#) ([sha4\\_context](#) \*ctx, unsigned char data[128])
- void [sha4\\_update](#) ([sha4\\_context](#) \*ctx, unsigned char \*input, int ilen)  
*SHA-512 process buffer.*
- void [sha4\\_finish](#) ([sha4\\_context](#) \*ctx, unsigned char output[64])  
*SHA-512 final digest.*
- void [sha4](#) (unsigned char \*input, int ilen, unsigned char output[64], int is384)  
*Output = SHA-512( input buffer ).*
- int [sha4\\_file](#) (char \*path, unsigned char output[64], int is384)  
*Output = SHA-512( file contents ).*
- void [sha4\\_hmac\\_starts](#) ([sha4\\_context](#) \*ctx, unsigned char \*key, int keylen, int is384)  
*SHA-512 HMAC context setup.*
- void [sha4\\_hmac\\_update](#) ([sha4\\_context](#) \*ctx, unsigned char \*input, int ilen)  
*SHA-512 HMAC process buffer.*
- void [sha4\\_hmac\\_finish](#) ([sha4\\_context](#) \*ctx, unsigned char output[64])

*SHA-512 HMAC final digest.*

- void `sha4_hmac` (unsigned char \*key, int keylen, unsigned char \*input, int ilen, unsigned char output[64], int is384)

*Output = HMAC-SHA-512( hmac key, input buffer ).*

- int `sha4_self_test` (int verbose)

*Checkup routine.*

## Variables

- static const unsigned int64 `K` [80]
- static const unsigned char `sha4_padding` [128]
- static unsigned char `sha4_test_buf` [3][113]
- static const int `sha4_test_buflen` [3]
- static const unsigned char `sha4_test_sum` [6][64]
- static unsigned char `sha4_hmac_test_key` [7][26]
- static const int `sha4_hmac_test_keylen` [7]
- static unsigned char `sha4_hmac_test_buf` [7][153]
- static const int `sha4_hmac_test_buflen` [7]
- static const unsigned char `sha4_hmac_test_sum` [14][64]

## 13.132.1 Define Documentation

**13.132.1.1** `#define F0(x, y, z) ((x & y) | (z & (x | y)))`

**13.132.1.2** `#define F1(x, y, z) (z ^ (x & (y ^ z)))`

**13.132.1.3** `#define GET_UINT64_BE(n, b, i)`

**Value:**

```
{
    (n) = ( (unsigned int64) (b)[(i)      ] << 56 ) \
    | ( (unsigned int64) (b)[(i) + 1] << 48 ) \
    | ( (unsigned int64) (b)[(i) + 2] << 40 ) \
    | ( (unsigned int64) (b)[(i) + 3] << 32 ) \
    | ( (unsigned int64) (b)[(i) + 4] << 24 ) \
    | ( (unsigned int64) (b)[(i) + 5] << 16 ) \
    | ( (unsigned int64) (b)[(i) + 6] <<  8 ) \
    | ( (unsigned int64) (b)[(i) + 7]      ); \
}
```

Definition at line 39 of file sha4.c.

Referenced by sha4\_process().

**13.132.1.4** `#define P(a, b, c, d, e, f, g, h, x, K)`

**Value:**



```

{
    temp1 = h + S3(e) + F1(e, f, g) + K + x;      \
    temp2 = S2(a) + F0(a, b, c);                  \
    d += temp1; h = temp1 + temp2;                 \
}

```

### 13.132.1.5 #define PUT\_UINT64\_BE(n, b, i)

**Value:**

```

{
    (b)[(i)    ] = (unsigned char) ( (n) >> 56 ); \
    (b)[(i) + 1] = (unsigned char) ( (n) >> 48 ); \
    (b)[(i) + 2] = (unsigned char) ( (n) >> 40 ); \
    (b)[(i) + 3] = (unsigned char) ( (n) >> 32 ); \
    (b)[(i) + 4] = (unsigned char) ( (n) >> 24 ); \
    (b)[(i) + 5] = (unsigned char) ( (n) >> 16 ); \
    (b)[(i) + 6] = (unsigned char) ( (n) >> 8  ); \
    (b)[(i) + 7] = (unsigned char) ( (n)          ); \
}

```

Definition at line 53 of file sha4.c.

Referenced by sha4\_finish().

### 13.132.1.6 #define ROTR(x, n) (SHR(x,n) | (x << (64 - n)))

### 13.132.1.7 #define S0(x) (ROTR(x, 1) ^ ROTR(x, 8) ^ SHR(x, 7))

### 13.132.1.8 #define S1(x) (ROTR(x,19) ^ ROTR(x,61) ^ SHR(x, 6))

### 13.132.1.9 #define S2(x) (ROTR(x,28) ^ ROTR(x,34) ^ ROTR(x,39))

### 13.132.1.10 #define S3(x) (ROTR(x,14) ^ ROTR(x,18) ^ ROTR(x,41))

### 13.132.1.11 #define SHR(x, n) (x >> n)

## 13.132.2 Function Documentation

### 13.132.2.1 void sha4 (unsigned char \*input, int ilen, unsigned char output[64], int is384)

Output = SHA-512( input buffer ).

**Parameters:**

*input* buffer holding the data  
*ilen* length of the input data  
*output* SHA-384/512 checksum result  
*is384* 0 = use SHA512, 1 = use SHA384

Definition at line 312 of file sha4.c.

References sha4\_finish(), sha4\_starts(), and sha4\_update().

Referenced by sha4\_hmac\_starts().

**13.132.2.2 int sha4\_file (char \* *path*, unsigned char *output*[64], int *is384*)**

Output = SHA-512( file contents ).

**Parameters:**

*path* input file name

*output* SHA-384/512 checksum result

*is384* 0 = use SHA512, 1 = use SHA384

**Returns:**

0 if successful, 1 if fopen failed, or 2 if fread failed

Definition at line 327 of file sha4.c.

References buf, f, sha4\_finish(), sha4\_starts(), and sha4\_update().

**13.132.2.3 void sha4\_finish (sha4\_context \* *ctx*, unsigned char *output*[64])**

SHA-512 final digest.

**Parameters:**

*ctx* SHA-512 context

*output* SHA-384/512 checksum result

Definition at line 276 of file sha4.c.

References int, int64, sha4\_context::is384, PUT\_UINT64\_BE, sha4\_padding, sha4\_update(), sha4\_context::state, and sha4\_context::total.

Referenced by sha4(), sha4\_file(), sha4\_hmac\_finish(), and sha4\_self\_test().

**13.132.2.4 void sha4\_hmac (unsigned char \* *key*, int *keylen*, unsigned char \* *input*, int *ilen*, unsigned char *output*[64], int *is384*)**

Output = HMAC-SHA-512( hmac key, input buffer ).

**Parameters:**

*key* HMAC secret key

*keylen* length of the HMAC key

*input* buffer holding the data

*ilen* length of the input data

*output* HMAC-SHA-384/512 result

*is384* 0 = use SHA512, 1 = use SHA384

Definition at line 419 of file sha4.c.

References sha4\_hmac\_finish(), sha4\_hmac\_starts(), and sha4\_hmac\_update().

### 13.132.2.5 void sha4\_hmac\_finish (sha4\_context \* *ctx*, unsigned char *output*[64])

SHA-512 HMAC final digest.

#### Parameters:

*ctx* HMAC context

*output* SHA-384/512 HMAC checksum result

Definition at line 399 of file sha4.c.

References sha4\_context::is384, sha4\_context::opad, sha4\_finish(), sha4\_starts(), and sha4\_update().

Referenced by sha4\_hmac(), and sha4\_self\_test().

### 13.132.2.6 void sha4\_hmac\_starts (sha4\_context \* *ctx*, unsigned char \* *key*, int *keylen*, int *is384*)

SHA-512 HMAC context setup.

#### Parameters:

*ctx* HMAC context to be initialized

*is384* 0 = use SHA512, 1 = use SHA384

*key* HMAC secret key

*keylen* length of the HMAC key

Definition at line 359 of file sha4.c.

References sha4\_context::ipad, sha4\_context::opad, sha4(), sha4\_starts(), and sha4\_update().

Referenced by sha4\_hmac(), and sha4\_self\_test().

### 13.132.2.7 void sha4\_hmac\_update (sha4\_context \* *ctx*, unsigned char \* *input*, int *ilen*)

SHA-512 HMAC process buffer.

#### Parameters:

*ctx* HMAC context

*input* buffer holding the data

*ilen* length of the input data

Definition at line 390 of file sha4.c.

References sha4\_update().

Referenced by sha4\_hmac(), and sha4\_self\_test().

### 13.132.2.8 static void sha4\_process (sha4\_context \* *ctx*, unsigned char *data*[128]) [static]

Definition at line 149 of file sha4.c.

References F, GET\_UINT64\_BE, int64, K, P, S0, S1, and sha4\_context::state.

Referenced by sha4\_update().

### 13.132.2.9 `int sha4_self_test (int verbose)`

Checkup routine.

#### Returns:

0 if successful, or 1 if the test failed

Definition at line 654 of file sha4.c.

References `buf`, `sha4_finish()`, `sha4_hmac_finish()`, `sha4_hmac_starts()`, `sha4_hmac_test_buf`, `sha4_hmac_test_buflen`, `sha4_hmac_test_key`, `sha4_hmac_test_keylen`, `sha4_hmac_test_sum`, `sha4_hmac_update()`, `sha4_starts()`, `sha4_test_buf`, `sha4_test_buflen`, `sha4_test_sum`, and `sha4_update()`.

Referenced by `main()`.

### 13.132.2.10 `void sha4_starts (sha4_context * ctx, int is384)`

SHA-512 context setup.

#### Parameters:

*ctx* context to be initialized

*is384* 0 = use SHA512, 1 = use SHA384

Definition at line 116 of file sha4.c.

References `sha4_context::is384`, `sha4_context::state`, `sha4_context::total`, and `UL64`.

Referenced by `sha4()`, `sha4_file()`, `sha4_hmac_finish()`, `sha4_hmac_starts()`, and `sha4_self_test()`.

### 13.132.2.11 `void sha4_update (sha4_context * ctx, unsigned char * input, int ilen)`

SHA-512 process buffer.

#### Parameters:

*ctx* SHA-512 context

*input* buffer holding the data

*ilen* length of the input data

Definition at line 221 of file sha4.c.

References `sha4_context::buffer`, `int`, `int64`, `sha4_process()`, and `sha4_context::total`.

Referenced by `sha4()`, `sha4_file()`, `sha4_finish()`, `sha4_hmac_finish()`, `sha4_hmac_starts()`, `sha4_hmac_update()`, and `sha4_self_test()`.

## 13.132.3 Variable Documentation

### 13.132.3.1 `const unsigned int64 K[80] [static]`

Definition at line 69 of file sha4.c.

**Initial value:**

Referenced by sha4\_self\_test().

**Initial value:**

Referenced by sha4\_self\_test().

**Initial value:**

Generated on Wed Mar 17 15:31:55 2010 for Mobile-C by Doxygen

Definition at line 506 of file sha4.c.

Referenced by sha4\_self\_test().

### 13.132.3.5 `const int sha4_hmac_test_keylen[7] [static]`

**Initial value:**

```
{
    20, 4, 20, 25, 20, 131, 131
}
```

Definition at line 521 of file sha4.c.

Referenced by sha4\_self\_test().

### 13.132.3.6 `const unsigned char sha4_hmac_test_sum[14][64] [static]`

Definition at line 552 of file sha4.c.

Referenced by sha4\_self\_test().

### 13.132.3.7 `const unsigned char sha4_padding[128] [static]`

**Initial value:**

```
{
    0x80, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
}
```

Definition at line 261 of file sha4.c.

Referenced by sha4\_finish().

### 13.132.3.8 `unsigned char sha4_test_buf[3][113] [static]`

**Initial value:**

```
{
    { "abc" },
    { "abcdefghijklghicdefghijdefghijkefghijklfghijklmghijklmn"
      "hijklmnoijklmnopjklmnopqklmnopqrlmnopqrsmnopqrstnoprstu" },
    { "" }
}
```

Definition at line 437 of file sha4.c.

Referenced by sha4\_self\_test().

### 13.132.3.9 `const int sha4_test_buflen[3] [static]`

Initial value:

```
{  
    3, 112, 1000  
}
```

Definition at line 445 of file sha4.c.

Referenced by sha4\_self\_test().

### 13.132.3.10 `const unsigned char sha4_test_sum[6][64] [static]`

Definition at line 450 of file sha4.c.

Referenced by sha4\_self\_test().

### 13.133 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/library/ssl\_cli.c File Reference

```
#include "xyssl/config.h"
#include "xyssl/debug.h"
#include "xyssl/ssl.h"
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <time.h>
```

#### Functions

- static [int ssl\\_write\\_client\\_hello](#) (ssl\_context \*ssl)
- static [int ssl\\_parse\\_server\\_hello](#) (ssl\_context \*ssl)
- static [int ssl\\_parse\\_server\\_key\\_exchange](#) (ssl\_context \*ssl)
- static [int ssl\\_parse\\_certificate\\_request](#) (ssl\_context \*ssl)
- static [int ssl\\_parse\\_server\\_hello\\_done](#) (ssl\_context \*ssl)
- static [int ssl\\_write\\_client\\_key\\_exchange](#) (ssl\_context \*ssl)
- static [int ssl\\_write\\_certificate\\_verify](#) (ssl\_context \*ssl)
- [int ssl\\_handshake\\_client](#) (ssl\_context \*ssl)

#### 13.133.1 Function Documentation

##### 13.133.1.1 [int ssl\\_handshake\\_client](#) (ssl\_context \*ssl)

Definition at line 660 of file `ssl_cli.c`.

References `SSL_CERTIFICATE_REQUEST`, `SSL_CERTIFICATE_VERIFY`, `SSL_CLIENT_CERTIFICATE`, `SSL_CLIENT_CHANGE_CIPHER_SPEC`, `SSL_CLIENT_FINISHED`, `SSL_CLIENT_HELLO`, `SSL_CLIENT_KEY_EXCHANGE`, `SSL_DEBUG_MSG`, `SSL_FLUSH_BUFFERS`, `ssl_flush_output()`, `SSL_HANDSHAKE_OVER`, `SSL_HELLO_REQUEST`, `ssl_parse_certificate()`, `ssl_parse_certificate_request()`, `ssl_parse_change_cipher_spec()`, `ssl_parse_finished()`, `ssl_parse_server_hello()`, `ssl_parse_server_hello_done()`, `ssl_parse_server_key_exchange()`, `SSL_SERVER_CERTIFICATE`, `SSL_SERVER_CHANGE_CIPHER_SPEC`, `SSL_SERVER_FINISHED`, `SSL_SERVER_HELLO`, `SSL_SERVER_HELLO_DONE`, `SSL_SERVER_KEY_EXCHANGE`, `ssl_write_certificate()`, `ssl_write_certificate_verify()`, `ssl_write_change_cipher_spec()`, `ssl_write_client_hello()`, `ssl_write_client_key_exchange()`, `ssl_write_finished()`, `_ssl_context::state`, and `XYSSL_ERR_SSL_BAD_INPUT_DATA`.

Referenced by `ssl_handshake()`.

##### 13.133.1.2 [static int ssl\\_parse\\_certificate\\_request](#) (ssl\_context \*ssl) **[static]**

Definition at line 428 of file `ssl_cli.c`.

References `_ssl_context::client_auth`, `_ssl_context::in_msg`, `_ssl_context::in_msgtype`, `SSL_DEBUG_MSG`, `SSL_DEBUG_RET`, `SSL_HS_CERTIFICATE_REQUEST`, `SSL_MSG_HANDSHAKE`, `ssl_read_record()`, `_ssl_context::state`, and `XYSSL_ERR_SSL_UNEXPECTED_MESSAGE`.

Referenced by `ssl_handshake_client()`.



#### 13.133.1.3 static int ssl\_parse\_server\_hello (ssl\_context \* ssl) [static]

Definition at line 168 of file ssl\_cli.c.

References buf, \_ssl\_session::cipher, \_ssl\_context::ciphers, \_ssl\_session::id, \_ssl\_context::in\_hslen, \_ssl\_context::in\_msg, \_ssl\_context::in\_msgtype, \_ssl\_session::length, \_ssl\_context::minor\_ver, \_ssl\_context::randbytes, \_ssl\_context::resume, \_ssl\_context::session, SSL\_COMPRESS\_NULL, SSL\_DEBUG\_BUF, SSL\_DEBUG\_MSG, SSL\_DEBUG\_RET, ssl\_derive\_keys(), SSL\_HS\_SERVER\_HELLO, SSL\_MAJOR\_VERSION\_3, SSL\_MINOR\_VERSION\_0, SSL\_MINOR\_VERSION\_1, SSL\_MSG\_HANDSHAKE, ssl\_read\_record(), SSL\_SERVER\_CHANGE\_CIPHER\_SPEC, \_ssl\_session::start, \_ssl\_context::state, XYSSL\_ERR\_SSL\_BAD\_HS\_SERVER\_HELLO, and XYSSL\_ERR\_SSL\_UNEXPECTED\_MESSAGE.

Referenced by ssl\_handshake\_client().

#### 13.133.1.4 static int ssl\_parse\_server\_hello\_done (ssl\_context \* ssl) [static]

Definition at line 471 of file ssl\_cli.c.

References \_ssl\_context::client\_auth, \_ssl\_context::in\_hslen, \_ssl\_context::in\_msg, \_ssl\_context::in\_msgtype, SSL\_DEBUG\_MSG, SSL\_DEBUG\_RET, SSL\_HS\_SERVER\_HELLO\_DONE, SSL\_MSG\_HANDSHAKE, ssl\_read\_record(), \_ssl\_context::state, XYSSL\_ERR\_SSL\_BAD\_HS\_SERVER\_HELLO\_DONE, and XYSSL\_ERR\_SSL\_UNEXPECTED\_MESSAGE.

Referenced by ssl\_handshake\_client().

#### 13.133.1.5 static int ssl\_parse\_server\_key\_exchange (ssl\_context \* ssl) [static]

Definition at line 312 of file ssl\_cli.c.

References \_ssl\_session::cipher, \_ssl\_context::dhm\_ctx, dhm\_read\_params(), dhm\_context::G, dhm\_context::GY, \_ssl\_context::in\_hslen, \_ssl\_context::in\_msg, \_ssl\_context::in\_msgtype, dhm\_context::len, rsa\_context::len, md5(), md5\_finish(), md5\_starts(), md5\_update(), dhm\_context::P, \_ssl\_context::peer\_cert, \_ssl\_context::randbytes, x509\_cert::rsa, rsa\_pkcs1\_verify(), RSA\_PUBLIC, RSA\_RAW, \_ssl\_context::session, sha1(), sha1\_finish(), sha1\_starts(), sha1\_update(), SSL\_DEBUG\_BUF, SSL\_DEBUG\_MPI, SSL\_DEBUG\_MSG, SSL\_DEBUG\_RET, SSL\_EDH\_RSA\_AES\_256\_SHA, SSL\_EDH\_RSA\_DES\_168\_SHA, SSL\_HS\_SERVER\_KEY\_EXCHANGE, SSL\_MSG\_HANDSHAKE, ssl\_read\_record(), \_ssl\_context::state, XYSSL\_ERR\_SSL\_BAD\_HS\_SERVER\_KEY\_EXCHANGE, XYSSL\_ERR\_SSL\_FEATURE\_UNAVAILABLE, and XYSSL\_ERR\_SSL\_UNEXPECTED\_MESSAGE.

Referenced by ssl\_handshake\_client().

#### 13.133.1.6 static int ssl\_write\_certificate\_verify (ssl\_context \* ssl) [static]

Definition at line 604 of file ssl\_cli.c.

References \_ssl\_context::client\_auth, rsa\_context::len, \_ssl\_context::out\_msg, \_ssl\_context::out\_msglen, \_ssl\_context::out\_msgtype, \_ssl\_context::rsa\_key, rsa\_pkcs1\_sign(), RSA\_PRIVATE, RSA\_RAW, ssl\_calc\_verify(), SSL\_DEBUG\_MSG, SSL\_DEBUG\_RET, SSL\_HS\_CERTIFICATE\_VERIFY, SSL\_MSG\_HANDSHAKE, ssl\_write\_record(), \_ssl\_context::state, and XYSSL\_ERR\_SSL\_PRIVATE\_KEY\_REQUIRED.

Referenced by ssl\_handshake\_client().

**13.133.1.7 static int ssl\_write\_client\_hello (ssl\_context \* ssl) [static]**

Definition at line 33 of file ssl\_cli.c.

References buf, \_ssl\_context::ciphers, \_ssl\_context::f\_rng, \_ssl\_context::hostname, \_ssl\_context::hostname\_len, \_ssl\_session::id, \_ssl\_session::length, \_ssl\_context::major\_ver, \_ssl\_context::max\_major\_ver, \_ssl\_context::max\_minor\_ver, \_ssl\_context::minor\_ver, \_ssl\_context::out\_msg, \_ssl\_context::out\_msglen, \_ssl\_context::out\_msgtype, \_ssl\_context::p\_rng, \_ssl\_context::randbytes, \_ssl\_context::resume, \_ssl\_context::session, SSL\_COMPRESS\_NULL, SSL\_DEBUG\_BUF, SSL\_DEBUG\_MSG, SSL\_DEBUG\_RET, SSL\_HS\_CLIENT\_HELLO, SSL\_MAJOR\_VERSION\_3, SSL\_MINOR\_VERSION\_0, SSL\_MINOR\_VERSION\_1, SSL\_MSG\_HANDSHAKE, ssl\_write\_record(), \_ssl\_session::start, \_ssl\_context::state, \_ssl\_context::timeout, TLS\_EXT\_SERVERNAME, and TLS\_EXT\_SERVERNAME\_HOSTNAME.

Referenced by ssl\_handshake\_client().

**13.133.1.8 static int ssl\_write\_client\_key\_exchange (ssl\_context \* ssl) [static]**

Definition at line 506 of file ssl\_cli.c.

References \_ssl\_session::cipher, dhm\_calc\_secret(), \_ssl\_context::dhm\_ctx, dhm\_make\_public(), \_ssl\_context::f\_rng, dhm\_context::GX, dhm\_context::K, rsa\_context::len, dhm\_context::len, \_ssl\_context::max\_major\_ver, \_ssl\_context::max\_minor\_ver, \_ssl\_context::minor\_ver, \_ssl\_context::out\_msg, \_ssl\_context::out\_msglen, \_ssl\_context::out\_msgtype, \_ssl\_context::p\_rng, \_ssl\_context::peer\_cert, \_ssl\_context::pmslen, \_ssl\_context::premaster, x509\_cert::rsa, rsa\_pkcs1\_encrypt(), RSA\_PUBLIC, \_ssl\_context::session, SSL\_DEBUG\_MPI, SSL\_DEBUG\_MSG, SSL\_DEBUG\_RET, ssl\_derive\_keys(), SSL\_EDH\_RSA\_AES\_256\_SHA, SSL\_EDH\_RSA\_DES\_168\_SHA, SSL\_HS\_CLIENT\_KEY\_EXCHANGE, SSL\_MINOR\_VERSION\_0, SSL\_MSG\_HANDSHAKE, ssl\_write\_record(), \_ssl\_context::state, dhm\_context::X, and XYSSL\_ERR\_SSL\_FEATURE\_UNAVAILABLE.

Referenced by ssl\_handshake\_client().

## 13.134 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/library/ssl\_srv.c File Reference

```
#include "xyssl/config.h"
#include "xyssl/debug.h"
#include "xyssl/ssl.h"
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <time.h>
```

### Functions

- static int [ssl\\_parse\\_client\\_hello](#) (ssl\_context \*ssl)
- static int [ssl\\_write\\_server\\_hello](#) (ssl\_context \*ssl)
- static int [ssl\\_write\\_certificate\\_request](#) (ssl\_context \*ssl)
- static int [ssl\\_write\\_server\\_key\\_exchange](#) (ssl\_context \*ssl)
- static int [ssl\\_write\\_server\\_hello\\_done](#) (ssl\_context \*ssl)
- static int [ssl\\_parse\\_client\\_key\\_exchange](#) (ssl\_context \*ssl)
- static int [ssl\\_parse\\_certificate\\_verify](#) (ssl\_context \*ssl)
- int [ssl\\_handshake\\_server](#) (ssl\_context \*ssl)

### 13.134.1 Function Documentation

#### 13.134.1.1 int [ssl\\_handshake\\_server](#) (ssl\_context \* ssl)

Definition at line 819 of file `ssl_srv.c`.

References `SSL_CERTIFICATE_REQUEST`, `SSL_CERTIFICATE_VERIFY`, `SSL_CLIENT_CERTIFICATE`, `SSL_CLIENT_CHANGE_CIPHER_SPEC`, `SSL_CLIENT_FINISHED`, `SSL_CLIENT_HELLO`, `SSL_CLIENT_KEY_EXCHANGE`, `SSL_DEBUG_MSG`, `SSL_FLUSH_BUFFERS`, `ssl_flush_output()`, `SSL_HANDSHAKE_OVER`, `SSL_HELLO_REQUEST`, `ssl_parse_certificate()`, `ssl_parse_certificate_verify()`, `ssl_parse_change_cipher_spec()`, `ssl_parse_client_hello()`, `ssl_parse_client_key_exchange()`, `ssl_parse_finished()`, `SSL_SERVER_CERTIFICATE`, `SSL_SERVER_CHANGE_CIPHER_SPEC`, `SSL_SERVER_FINISHED`, `SSL_SERVER_HELLO`, `SSL_SERVER_HELLO_DONE`, `SSL_SERVER_KEY_EXCHANGE`, `ssl_write_certificate()`, `ssl_write_certificate_request()`, `ssl_write_change_cipher_spec()`, `ssl_write_finished()`, `ssl_write_server_hello()`, `ssl_write_server_hello_done()`, `ssl_write_server_key_exchange()`, `_ssl_context::state`, and `XYSSL_ERR_SSL_BAD_INPUT_DATA`.

Referenced by `ssl_handshake()`.

#### 13.134.1.2 static int [ssl\\_parse\\_certificate\\_verify](#) (ssl\_context \* ssl) [static]

Definition at line 758 of file `ssl_srv.c`.

References `_ssl_context::in_hrlen`, `_ssl_context::in_msg`, `_ssl_context::in_msgtype`, `rsa_context::len`, `_ssl_context::peer_cert`, `_x509_cert::rsa`, `rsa_pkcs1_verify()`, `RSA_PUBLIC`, `RSA_RAW`, `ssl_`

calc\_verify(), SSL\_DEBUG\_MSG, SSL\_DEBUG\_RET, SSL\_HS\_CERTIFICATE\_VERIFY, SSL\_MSG\_HANDSHAKE, ssl\_read\_record(), \_ssl\_context::state, and XYSSL\_ERR\_SSL\_BAD\_HS\_CERTIFICATE\_VERIFY.

Referenced by ssl\_handshake\_server().

#### 13.134.1.3 static int ssl\_parse\_client\_hello (ssl\_context \* ssl) [static]

Definition at line 33 of file ssl\_srv.c.

References buf, \_ssl\_session::cipher, \_ssl\_context::ciphers, \_ssl\_context::fin\_md5, \_ssl\_context::fin\_sha1, \_ssl\_session::id, \_ssl\_context::in\_hdr, \_ssl\_context::in\_left, \_ssl\_context::in\_msg, \_ssl\_session::length, \_ssl\_context::major\_ver, \_ssl\_context::max\_major\_ver, \_ssl\_context::max\_minor\_ver, md5\_update(), \_ssl\_context::minor\_ver, \_ssl\_context::randbytes, \_ssl\_context::session, sha1\_update(), SSL\_DEBUG\_BUF, SSL\_DEBUG\_MSG, SSL\_DEBUG\_RET, ssl\_fetch\_input(), SSL\_HS\_CLIENT\_HELLO, SSL\_MAJOR\_VERSION\_3, SSL\_MINOR\_VERSION\_1, SSL\_MSG\_HANDSHAKE, \_ssl\_context::state, XYSSL\_ERR\_SSL\_BAD\_HS\_CLIENT\_HELLO, and XYSSL\_ERR\_SSL\_NO\_CIPHER\_CHOSEN.

Referenced by ssl\_handshake\_server().

#### 13.134.1.4 static int ssl\_parse\_client\_key\_exchange (ssl\_context \* ssl) [static]

Definition at line 634 of file ssl\_srv.c.

References \_ssl\_session::cipher, dhm\_calc\_secret(), \_ssl\_context::dhm\_ctx, dhm\_read\_public(), \_ssl\_context::f\_rng, dhm\_context::GY, \_ssl\_context::in\_hslen, \_ssl\_context::in\_msg, \_ssl\_context::in\_msgtype, dhm\_context::K, rsa\_context::len, dhm\_context::len, \_ssl\_context::max\_major\_ver, \_ssl\_context::max\_minor\_ver, \_ssl\_context::minor\_ver, \_ssl\_context::p\_rng, \_ssl\_context::pmslen, \_ssl\_context::premaster, \_ssl\_context::rsa\_key, rsa\_pkcs1\_decrypt(), RSA\_PRIVATE, \_ssl\_context::s\_set, \_ssl\_context::session, SSL\_DEBUG\_MPI, SSL\_DEBUG\_MSG, SSL\_DEBUG\_RET, ssl\_derive\_keys(), SSL\_EDH\_RSA\_AES\_256\_SHA, SSL\_EDH\_RSA\_DES\_168\_SHA, SSL\_HS\_CLIENT\_KEY\_EXCHANGE, SSL\_MINOR\_VERSION\_0, SSL\_MSG\_HANDSHAKE, ssl\_read\_record(), \_ssl\_context::state, XYSSL\_ERR\_SSL\_BAD\_HS\_CLIENT\_KEY\_EXCHANGE, and XYSSL\_ERR\_SSL\_FEATURE\_UNAVAILABLE.

Referenced by ssl\_handshake\_server().

#### 13.134.1.5 static int ssl\_write\_certificate\_request (ssl\_context \* ssl) [static]

Definition at line 447 of file ssl\_srv.c.

References \_ssl\_context::authmode, buf, \_ssl\_context::ca\_chain, \_x509\_buf::len, \_x509\_cert::next, \_ssl\_context::out\_msg, \_ssl\_context::out\_msglen, \_ssl\_context::out\_msgtype, \_x509\_buf::p, SSL\_DEBUG\_BUF, SSL\_DEBUG\_MSG, SSL\_HS\_CERTIFICATE\_REQUEST, SSL\_MSG\_HANDSHAKE, SSL\_VERIFY\_NONE, ssl\_write\_record(), \_ssl\_context::state, and \_x509\_cert::subject\_raw.

Referenced by ssl\_handshake\_server().

#### 13.134.1.6 static int ssl\_write\_server\_hello (ssl\_context \* ssl) [static]

Definition at line 349 of file ssl\_srv.c.

References buf, \_ssl\_session::cipher, \_ssl\_context::f\_rng, \_ssl\_session::id, \_ssl\_session::length, \_ssl\_context::major\_ver, \_ssl\_context::minor\_ver, \_ssl\_context::out\_msg, \_ssl\_context::out\_msglen, \_ssl\_context::out\_msgtype, \_ssl\_context::p\_rng, \_ssl\_context::randbytes, \_ssl\_context::resume, \_ssl\_

context::s\_get, \_ssl\_context::session, SSL\_COMPRESS\_NULL, SSL\_DEBUG\_BUF, SSL\_DEBUG\_MSG, ssl\_derive\_keys(), SSL\_HS\_SERVER\_HELLO, SSL\_MSG\_HANDSHAKE, SSL\_SERVER\_CHANGE\_CIPHER\_SPEC, ssl\_write\_record(), and \_ssl\_context::state.

Referenced by ssl\_handshake\_server().

#### 13.134.1.7 static int ssl\_write\_server\_hello\_done(ssl\_context \*ssl) [static]

Definition at line 611 of file ssl\_srv.c.

References \_ssl\_context::out\_msg, \_ssl\_context::out\_msglen, \_ssl\_context::out\_msgtype, SSL\_DEBUG\_MSG, SSL\_DEBUG\_RET, SSL\_HS\_SERVER\_HELLO\_DONE, SSL\_MSG\_HANDSHAKE, ssl\_write\_record(), and \_ssl\_context::state.

Referenced by ssl\_handshake\_server().

#### 13.134.1.8 static int ssl\_write\_server\_key\_exchange(ssl\_context \*ssl) [static]

Definition at line 512 of file ssl\_srv.c.

References \_ssl\_session::cipher, \_ssl\_context::dhm\_ctx, dhm\_make\_params(), \_ssl\_context::f\_rng, dhm\_context::G, dhm\_context::GX, rsa\_context::len, md5(), md5\_finish(), md5\_starts(), md5\_update(), \_ssl\_context::out\_msg, \_ssl\_context::out\_msglen, \_ssl\_context::out\_msgtype, dhm\_context::P, \_ssl\_context::p\_rng, \_ssl\_context::randbytes, \_ssl\_context::rsa\_key, rsa\_pkcs1\_sign(), RSA\_PRIVATE, RSA\_RAW, \_ssl\_context::session, sha1(), sha1\_finish(), sha1\_starts(), sha1\_update(), SSL\_DEBUG\_BUF, SSL\_DEBUG\_MPI, SSL\_DEBUG\_MSG, SSL\_DEBUG\_RET, SSL\_EDH\_RSA\_AES\_256\_SHA, SSL\_EDH\_RSA\_DES\_168\_SHA, SSL\_HS\_SERVER\_KEY\_EXCHANGE, SSL\_MSG\_HANDSHAKE, ssl\_write\_record(), \_ssl\_context::state, dhm\_context::X, and XYSSL\_ERR\_SSL\_FEATURE\_UNAVAILABLE.

Referenced by ssl\_handshake\_server().

### 13.135 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/library/ssl\_tls.c File Reference

```
#include "xyssl/config.h"
#include "xyssl/aes.h"
#include "xyssl/arc4.h"
#include "xyssl/des.h"
#include "xyssl/debug.h"
#include "xyssl/ssl.h"
#include <string.h>
#include <stdlib.h>
#include <time.h>
```

#### Functions

- static [int](#) [tls1\\_prf](#) (unsigned char \*secret, [int](#) slen, char \*label, unsigned char \*random, [int](#) rlen, unsigned char \*dstbuf, [int](#) dlen)
- [int](#) [ssl\\_derive\\_keys](#) ([ssl\\_context](#) \*ssl)
- void [ssl\\_calc\\_verify](#) ([ssl\\_context](#) \*ssl, unsigned char hash[36])
- static void [ssl\\_mac\\_md5](#) (unsigned char \*secret, unsigned char \*buf, [int](#) len, unsigned char \*ctr, [int](#) type)
- static void [ssl\\_mac\\_sha1](#) (unsigned char \*secret, unsigned char \*buf, [int](#) len, unsigned char \*ctr, [int](#) type)
- static [int](#) [ssl\\_encrypt\\_buf](#) ([ssl\\_context](#) \*ssl)
- static [int](#) [ssl\\_decrypt\\_buf](#) ([ssl\\_context](#) \*ssl)
- [int](#) [ssl\\_fetch\\_input](#) ([ssl\\_context](#) \*ssl, [int](#) nb\_want)
- [int](#) [ssl\\_flush\\_output](#) ([ssl\\_context](#) \*ssl)
- [int](#) [ssl\\_write\\_record](#) ([ssl\\_context](#) \*ssl)
- [int](#) [ssl\\_read\\_record](#) ([ssl\\_context](#) \*ssl)
- [int](#) [ssl\\_write\\_certificate](#) ([ssl\\_context](#) \*ssl)
- [int](#) [ssl\\_parse\\_certificate](#) ([ssl\\_context](#) \*ssl)
- [int](#) [ssl\\_write\\_change\\_cipher\\_spec](#) ([ssl\\_context](#) \*ssl)
- [int](#) [ssl\\_parse\\_change\\_cipher\\_spec](#) ([ssl\\_context](#) \*ssl)
- static void [ssl\\_calc\\_finished](#) ([ssl\\_context](#) \*ssl, unsigned char \*buf, [int](#) from, [md5\\_context](#) \*md5, [sha1\\_context](#) \*sha1)
- [int](#) [ssl\\_write\\_finished](#) ([ssl\\_context](#) \*ssl)
- [int](#) [ssl\\_parse\\_finished](#) ([ssl\\_context](#) \*ssl)
- [int](#) [ssl\\_init](#) ([ssl\\_context](#) \*ssl)

*Initialize an SSL context.*

- void [ssl\\_set\\_endpoint](#) ([ssl\\_context](#) \*ssl, [int](#) endpoint)

*Set the current endpoint type.*

- void [ssl\\_set\\_authmode](#) ([ssl\\_context](#) \*ssl, [int](#) authmode)

*Set the certificate verification mode.*

- void `ssl_set_rng` (`ssl_context` \*ssl, `int`(\*f\_rng)(void \*), void \*p\_rng)  
*Set the random number generator callback.*
- void `ssl_set_dbg` (`ssl_context` \*ssl, void(\*f\_dbg)(void \*, `int`, char \*), void \*p\_dbg)  
*Set the debug callback.*
- void `ssl_set_bio` (`ssl_context` \*ssl, `int`(\*f\_recv)(void \*, unsigned char \*, `int`), void \*p\_recv, `int`(\*f\_send)(void \*, unsigned char \*, `int`), void \*p\_send)  
*Set the underlying BIO read and write callbacks.*
- void `ssl_set_scb` (`ssl_context` \*ssl, `int`(\*s\_get)(`ssl_context` \*), `int`(\*s\_set)(`ssl_context` \*))  
*Set the session callbacks (server-side only).*
- void `ssl_set_session` (`ssl_context` \*ssl, `int` resume, `int` timeout, `ssl_session` \*session)  
*Set the session resuming flag, timeout and data.*
- void `ssl_set_ciphers` (`ssl_context` \*ssl, `int` \*ciphers)  
*Set the list of allowed ciphersuites.*
- void `ssl_set_ca_chain` (`ssl_context` \*ssl, `x509_cert` \*ca\_chain, char \*peer\_cn)  
*Set the data required to verify peer certificate.*
- void `ssl_set_own_cert` (`ssl_context` \*ssl, `x509_cert` \*own\_cert, `rsa_context` \*rsa\_key)  
*Set own certificate and private key.*
- `int` `ssl_set_dh_param` (`ssl_context` \*ssl, char \*dhm\_P, char \*dhm\_G)  
*Set the Diffie-Hellman public P and G values, read as hexadecimal strings (server-side only).*
- `int` `ssl_set_hostname` (`ssl_context` \*ssl, char \*hostname)  
*Set hostname for ServerName TLS Extension.*
- `int` `ssl_get_bytes_avail` (`ssl_context` \*ssl)  
*Return the number of data bytes available to read.*
- `int` `ssl_get_verify_result` (`ssl_context` \*ssl)  
*Return the result of the certificate verification.*
- char \* `ssl_get_cipher` (`ssl_context` \*ssl)  
*Return the name of the current cipher.*
- `int` `ssl_handshake` (`ssl_context` \*ssl)  
*Perform the SSL handshake.*
- `int` `ssl_read` (`ssl_context` \*ssl, unsigned char \*buf, `int` len)  
*Read at most 'len' application data bytes.*
- `int` `ssl_write` (`ssl_context` \*ssl, unsigned char \*buf, `int` len)  
*Write exactly 'len' application data bytes.*
- `int` `ssl_close_notify` (`ssl_context` \*ssl)

*Notify the peer that the connection is being closed.*

- void `ssl_free` (`ssl_context` \*`ssl`)

*Free an SSL context.*

## Variables

- int `ssl_default_ciphers` []

### 13.135.1 Function Documentation

#### 13.135.1.1 static void `ssl_calc_finished` (`ssl_context` \*`ssl`, unsigned char \*`buf`, int `from`, `md5_context` \*`md5`, `sha1_context` \*`sha1`) [static]

Definition at line 1344 of file `ssl_tls.c`.

References `_ssl_session::master`, `md5_finish()`, `md5_starts()`, `md5_update()`, `_ssl_context::minor_ver`, `_ssl_context::session`, `sha1_finish()`, `sha1_starts()`, `sha1_update()`, `SSL_DEBUG_BUF`, `SSL_DEBUG_MSG`, `SSL_IS_CLIENT`, `SSL_MINOR_VERSION_0`, `sha1_context::state`, `md5_context::state`, and `tls1_prf()`.

Referenced by `ssl_parse_finished()`, and `ssl_write_finished()`.

#### 13.135.1.2 void `ssl_calc_verify` (`ssl_context` \*`ssl`, unsigned char `hash`[36])

Definition at line 335 of file `ssl_tls.c`.

References `_ssl_context::fin_md5`, `_ssl_context::fin_sha1`, `_ssl_session::master`, `md5()`, `md5_finish()`, `md5_starts()`, `md5_update()`, `_ssl_context::minor_ver`, `_ssl_context::session`, `sha1()`, `sha1_finish()`, `sha1_starts()`, `sha1_update()`, `SSL_DEBUG_BUF`, `SSL_DEBUG_MSG`, and `SSL_MINOR_VERSION_0`.

Referenced by `ssl_parse_certificate_verify()`, and `ssl_write_certificate_verify()`.

#### 13.135.1.3 int `ssl_close_notify` (`ssl_context` \*`ssl`)

*Notify the peer that the connection is being closed.*

Definition at line 1904 of file `ssl_tls.c`.

References `_ssl_context::out_msg`, `_ssl_context::out_msglen`, `_ssl_context::out_msgtype`, `SSL_ALERT_CLOSE_NOTIFY`, `SSL_ALERT_WARNING`, `SSL_DEBUG_MSG`, `SSL_DEBUG_RET`, `ssl_flush_output()`, `SSL_HANDSHAKE_OVER`, `SSL_MSG_ALERT`, `ssl_write_record()`, and `_ssl_context::state`.

Referenced by `main()`, and `ssl_test()`.

#### 13.135.1.4 static int `ssl_decrypt_buf` (`ssl_context` \*`ssl`) [static]

Definition at line 555 of file `ssl_tls.c`.

References `aes_crypt_cbc()`, `AES_DECRYPT`, `arc4_crypt()`, `_ssl_context::ctx_dec`, `des3_crypt_cbc()`, `DES_DECRYPT`, `_ssl_context::in_ctr`, `_ssl_context::in_hdr`, `_ssl_context::in_msg`, `_ssl_context::in_msglen`, `_ssl_context::in_msgtype`, `_ssl_context::iv_dec`, `_ssl_context::ivlen`, `_ssl_context::mac_dec`, `_ssl_context::maclen`, `md5_hmac()`, `_ssl_context::minlen`, `_ssl_context::minor_ver`, `_ssl_context::nb_`



zero, sha1\_hmac(), SSL\_DEBUG\_BUF, SSL\_DEBUG\_MSG, ssl\_mac\_md5(), ssl\_mac\_sha1(), SSL\_MINOR\_VERSION\_0, XYSSL\_ERR\_SSL\_FEATURE\_UNAVAILABLE, and XYSSL\_ERR\_SSL\_INVALID\_MAC.

Referenced by ssl\_read\_record().

#### **13.135.1.5 int ssl\_derive\_keys (ssl\_context \* ssl)**

Definition at line 106 of file ssl\_tls.c.

References aes\_setkey\_dec(), aes\_setkey\_enc(), arc4\_setup(), \_ssl\_session::cipher, \_ssl\_context::ctx\_dec, \_ssl\_context::ctx\_enc, des3\_set3key\_dec(), des3\_set3key\_enc(), \_ssl\_context::endpoint, \_ssl\_context::iv\_dec, \_ssl\_context::iv\_enc, \_ssl\_context::ivlen, \_ssl\_context::keylen, \_ssl\_context::mac\_dec, \_ssl\_context::mac\_enc, \_ssl\_context::maclen, \_ssl\_session::master, md5(), md5\_finish(), md5\_starts(), md5\_update(), \_ssl\_context::minlen, \_ssl\_context::minor\_ver, \_ssl\_context::pmslen, \_ssl\_context::premaster, \_ssl\_context::randbytes, \_ssl\_context::resume, \_ssl\_context::session, sha1(), sha1\_finish(), sha1\_starts(), sha1\_update(), SSL\_DEBUG\_BUF, SSL\_DEBUG\_MSG, SSL\_EDH\_RSA\_AES\_256\_SHA, SSL\_EDH\_RSA\_DES\_168\_SHA, ssl\_get\_cipher(), SSL\_IS\_CLIENT, SSL\_MINOR\_VERSION\_0, SSL\_RSA\_AES\_128\_SHA, SSL\_RSA\_AES\_256\_SHA, SSL\_RSA\_DES\_168\_SHA, SSL\_RSA\_RC4\_128\_MD5, SSL\_RSA\_RC4\_128\_SHA, tls1\_prf(), and XYSSL\_ERR\_SSL\_FEATURE\_UNAVAILABLE.

Referenced by ssl\_parse\_client\_key\_exchange(), ssl\_parse\_server\_hello(), ssl\_write\_client\_key\_exchange(), and ssl\_write\_server\_hello().

#### **13.135.1.6 static int ssl\_encrypt\_buf (ssl\_context \* ssl) [static]**

Definition at line 448 of file ssl\_tls.c.

References aes\_crypt\_cbc(), AES\_ENCRYPT, arc4\_crypt(), \_ssl\_context::ctx\_enc, des3\_crypt\_cbc(), DES\_ENCRYPT, \_ssl\_context::iv\_enc, \_ssl\_context::ivlen, \_ssl\_context::mac\_enc, \_ssl\_context::maclen, md5\_hmac(), \_ssl\_context::minor\_ver, \_ssl\_context::out\_ctr, \_ssl\_context::out\_msg, \_ssl\_context::out\_msglen, \_ssl\_context::out\_msgtype, sha1\_hmac(), SSL\_DEBUG\_BUF, SSL\_DEBUG\_MSG, ssl\_mac\_md5(), ssl\_mac\_sha1(), SSL\_MINOR\_VERSION\_0, and XYSSL\_ERR\_SSL\_FEATURE\_UNAVAILABLE.

Referenced by ssl\_write\_record().

#### **13.135.1.7 int ssl\_fetch\_input (ssl\_context \* ssl, int nb\_want)**

Definition at line 727 of file ssl\_tls.c.

References \_ssl\_context::f\_recv, \_ssl\_context::in\_hdr, \_ssl\_context::in\_left, \_ssl\_context::p\_recv, SSL\_DEBUG\_MSG, and SSL\_DEBUG\_RET.

Referenced by ssl\_parse\_client\_hello(), and ssl\_read\_record().

#### **13.135.1.8 int ssl\_flush\_output (ssl\_context \* ssl)**

Definition at line 756 of file ssl\_tls.c.

References buf, \_ssl\_context::f\_send, \_ssl\_context::out\_hdr, \_ssl\_context::out\_left, \_ssl\_context::out\_msglen, \_ssl\_context::p\_send, SSL\_DEBUG\_MSG, and SSL\_DEBUG\_RET.

Referenced by ssl\_close\_notify(), ssl\_handshake\_client(), ssl\_handshake\_server(), ssl\_write(), and ssl\_write\_record().

**13.135.1.9 void ssl\_free (ssl\_context \* ssl)**

Free an SSL context.

Definition at line 1938 of file ssl\_tls.c.

References `_ssl_context::dhm_ctx`, `dhm_free()`, `_ssl_context::hostname`, `_ssl_context::hostname_len`, `_ssl_context::in_ctr`, `_ssl_context::out_ctr`, `_ssl_context::peer_cert`, `SSL_BUFFER_LEN`, `SSL_DEBUG_MSG`, and `x509_free()`.

Referenced by `main()`, and `ssl_test()`.

**13.135.1.10 int ssl\_get\_bytes\_avail (ssl\_context \* ssl)**

Return the number of data bytes available to read.

**Parameters:**

*ssl* SSL context

**Returns:**

how many bytes are available in the read buffer

Definition at line 1691 of file ssl\_tls.c.

References `_ssl_context::in_msglen`, and `_ssl_context::in_offt`.

**13.135.1.11 char\* ssl\_get\_cipher (ssl\_context \* ssl)**

Return the name of the current cipher.

**Parameters:**

*ssl* SSL context

**Returns:**

a string containing the cipher name

Definition at line 1701 of file ssl\_tls.c.

References `_ssl_session::cipher`, `_ssl_context::session`, `SSL_EDH_RSA_AES_256_SHA`, `SSL_EDH_RSA_DES_168_SHA`, `SSL_RSA_AES_128_SHA`, `SSL_RSA_AES_256_SHA`, `SSL_RSA_DES_168_SHA`, `SSL_RSA_RC4_128_MD5`, and `SSL_RSA_RC4_128_SHA`.

Referenced by `main()`, and `ssl_derive_keys()`.

**13.135.1.12 int ssl\_get\_verify\_result (ssl\_context \* ssl)**

Return the result of the certificate verification.

**Parameters:**

*ssl* SSL context

**Returns:**

0 if successful, or a combination of: BADCERT\_EXPIRED BADCERT\_REVOKED BADCERT\_CN\_MISMATCH BADCERT\_NOT\_TRUSTED

Definition at line 1696 of file ssl\_tls.c.

References `_ssl_context::verify_result`.

Referenced by `main()`.

**13.135.1.13 int ssl\_handshake (ssl\_context \* ssl)**

Perform the SSL handshake.

**Parameters:**

*ssl* SSL context

**Returns:**

0 if successful, XYSSL\_ERR\_NET\_TRY\_AGAIN, or a specific SSL error code.

Definition at line 1767 of file ssl\_tls.c.

References `_ssl_context::endpoint`, `SSL_DEBUG_MSG`, `ssl_handshake_client()`, `ssl_handshake_server()`, `SSL_IS_CLIENT`, `SSL_IS_SERVER`, and `XYSSL_ERR_SSL_FEATURE_UNAVAILABLE`.

Referenced by `main()`, `ssl_read()`, and `ssl_write()`.

**13.135.1.14 int ssl\_init (ssl\_context \* ssl)**

Initialize an SSL context.

**Parameters:**

*ssl* SSL context

**Returns:**

0 if successful, or 1 if memory allocation failed

Definition at line 1542 of file ssl\_tls.c.

References `_ssl_context::fin_md5`, `_ssl_context::fin_sha1`, `_ssl_context::hostname`, `_ssl_context::hostname_len`, `_ssl_context::in_ctr`, `_ssl_context::in_hdr`, `_ssl_context::in_msg`, `md5_starts()`, `_ssl_context::out_ctr`, `_ssl_context::out_hdr`, `_ssl_context::out_msg`, `sha1_starts()`, `SSL_BUFFER_LEN`, and `SSL_DEBUG_MSG`.

Referenced by `main()`, and `ssl_test()`.

**13.135.1.15 static void ssl\_mac\_md5 (unsigned char \* secret, unsigned char \* buf, int len, unsigned char \* ctr, int type) [static]**

Definition at line 387 of file ssl\_tls.c.

References `md5()`, `md5_finish()`, `md5_starts()`, and `md5_update()`.

Referenced by `ssl_decrypt_buf()`, and `ssl_encrypt_buf()`.

### 13.135.1.16 `static void ssl_mac_sha1 (unsigned char * secret, unsigned char * buf, int len, unsigned char * ctr, int type) [static]`

Definition at line 416 of file `ssl_tls.c`.

References `sha1()`, `sha1_finish()`, `sha1_starts()`, and `sha1_update()`.

Referenced by `ssl_decrypt_buf()`, and `ssl_encrypt_buf()`.

### 13.135.1.17 `int ssl_parse_certificate (ssl_context * ssl)`

Definition at line 1140 of file `ssl_tls.c`.

References `_ssl_context::authmode`, `_ssl_context::ca_chain`, `_ssl_context::endpoint`, `_ssl_context::in_hrlen`, `_ssl_context::in_msg`, `_ssl_context::in_msglen`, `_ssl_context::in_msgtype`, `int`, `_ssl_context::minor_ver`, `_ssl_context::peer_cert`, `_ssl_context::peer_cn`, `SSL_ALERT_NO_CERTIFICATE`, `SSL_ALERT_WARNING`, `SSL_DEBUG_CRT`, `SSL_DEBUG_MSG`, `SSL_DEBUG_RET`, `SSL_HS_CERTIFICATE`, `SSL_IS_SERVER`, `SSL_MINOR_VERSION_0`, `SSL_MSG_ALERT`, `SSL_MSG_HANDSHAKE`, `ssl_read_record()`, `SSL_VERIFY_NONE`, `SSL_VERIFY_OPTIONAL`, `SSL_VERIFY_REQUIRED`, `_ssl_context::state`, `_ssl_context::verify_result`, `x509parse_cert()`, `x509parse_verify()`, `XYSSL_ERR_SSL_BAD_HS_CERTIFICATE`, `XYSSL_ERR_SSL_CA_CHAIN_REQUIRED`, `XYSSL_ERR_SSL_NO_CLIENT_CERTIFICATE`, and `XYSSL_ERR_SSL_UNEXPECTED_MESSAGE`.

Referenced by `ssl_handshake_client()`, and `ssl_handshake_server()`.

### 13.135.1.18 `int ssl_parse_change_cipher_spec (ssl_context * ssl)`

Definition at line 1311 of file `ssl_tls.c`.

References `_ssl_context::do_crypt`, `_ssl_context::in_msg`, `_ssl_context::in_msglen`, `_ssl_context::in_msgtype`, `SSL_DEBUG_MSG`, `SSL_DEBUG_RET`, `SSL_MSG_CHANGE_CIPHER_SPEC`, `ssl_read_record()`, `_ssl_context::state`, `XYSSL_ERR_SSL_BAD_HS_CHANGE_CIPHER_SPEC`, and `XYSSL_ERR_SSL_UNEXPECTED_MESSAGE`.

Referenced by `ssl_handshake_client()`, and `ssl_handshake_server()`.

### 13.135.1.19 `int ssl_parse_finished (ssl_context * ssl)`

Definition at line 1480 of file `ssl_tls.c`.

References `buf`, `_ssl_context::do_crypt`, `_ssl_context::endpoint`, `_ssl_context::fin_md5`, `_ssl_context::fin_sha1`, `_ssl_context::in_hrlen`, `_ssl_context::in_msg`, `_ssl_context::in_msgtype`, `md5()`, `_ssl_context::minor_ver`, `_ssl_context::resume`, `sha1()`, `ssl_calc_finished()`, `SSL_CLIENT_CHANGE_CIPHER_SPEC`, `SSL_DEBUG_MSG`, `SSL_DEBUG_RET`, `SSL_HANDSHAKE_OVER`, `SSL_HS_FINISHED`, `SSL_IS_CLIENT`, `SSL_IS_SERVER`, `SSL_MINOR_VERSION_0`, `SSL_MSG_HANDSHAKE`, `ssl_read_record()`, `_ssl_context::state`, `XYSSL_ERR_SSL_BAD_HS_FINISHED`, and `XYSSL_ERR_SSL_UNEXPECTED_MESSAGE`.

Referenced by `ssl_handshake_client()`, and `ssl_handshake_server()`.

### 13.135.1.20 `int ssl_read (ssl_context * ssl, unsigned char * buf, int len)`

Read at most 'len' application data bytes.

**Parameters:**

*ssl* SSL context  
*buf* buffer that will hold the data  
*len* how many bytes must be read

**Returns:**

This function returns the number of bytes read, or a negative error code.

Definition at line 1791 of file ssl\_tls.c.

References `_ssl_context::in_msg`, `_ssl_context::in_msglen`, `_ssl_context::in_msgtype`, `_ssl_context::in_offt`, `SSL_DEBUG_MSG`, `SSL_DEBUG_RET`, `ssl_handshake()`, `SSL_HANDSHAKE_OVER`, `SSL_MSG_APPLICATION_DATA`, `ssl_read_record()`, `_ssl_context::state`, and `XYSSL_ERR_SSL_UNEXPECTED_MESSAGE`.

Referenced by `main()`, and `ssl_test()`.

**13.135.1.21 int ssl\_read\_record (ssl\_context \* ssl)**

Definition at line 842 of file ssl\_tls.c.

References `_ssl_context::do_crypt`, `_ssl_context::fin_md5`, `_ssl_context::fin_sha1`, `_ssl_context::in_hdr`, `_ssl_context::in_hrlen`, `_ssl_context::in_left`, `_ssl_context::in_msg`, `_ssl_context::in_msglen`, `_ssl_context::in_msgtype`, `_ssl_context::major_ver`, `md5_update()`, `_ssl_context::minlen`, `_ssl_context::minor_ver`, `sha1_update()`, `SSL_ALERT_CLOSE_NOTIFY`, `SSL_ALERT_FATAL`, `SSL_ALERT_WARNING`, `SSL_DEBUG_BUF`, `SSL_DEBUG_MSG`, `SSL_DEBUG_RET`, `ssl_decrypt_buf()`, `ssl_fetch_input()`, `SSL_MAX_CONTENT_LEN`, `SSL_MINOR_VERSION_0`, `SSL_MINOR_VERSION_1`, `SSL_MSG_ALERT`, `SSL_MSG_HANDSHAKE`, `XYSSL_ERR_SSL_FATAL_ALERT_MESSAGE`, `XYSSL_ERR_SSL_INVALID_RECORD`, and `XYSSL_ERR_SSL_PEER_CLOSE_NOTIFY`.

Referenced by `ssl_parse_certificate()`, `ssl_parse_certificate_request()`, `ssl_parse_certificate_verify()`, `ssl_parse_change_cipher_spec()`, `ssl_parse_client_key_exchange()`, `ssl_parse_finished()`, `ssl_parse_server_hello()`, `ssl_parse_server_hello_done()`, `ssl_parse_server_key_exchange()`, and `ssl_read()`.

**13.135.1.22 void ssl\_set\_authmode (ssl\_context \* ssl, int authmode)**

Set the certificate verification mode.

**Parameters:**

*ssl* SSL context  
*mode* can be:

`SSL_VERIFY_NONE`: peer certificate is not checked (default), this is insecure and SHOULD be avoided.

`SSL_VERIFY_OPTIONAL`: peer certificate is checked, however the handshake continues even if verification failed; [ssl\\_get\\_verify\\_result\(\)](#) can be called after the handshake is complete.

`SSL_VERIFY_REQUIRED`: peer *must* present a valid certificate, handshake is aborted if verification failed.

Definition at line 1589 of file ssl\_tls.c.

References `_ssl_context::authmode`.

Referenced by `main()`, and `ssl_test()`.

**13.135.1.23** `void ssl_set_bio (ssl_context * ssl, int(*)(void *, unsigned char *, int) f_recv, void * p_recv, int(*)(void *, unsigned char *, int) f_send, void * p_send)`

Set the underlying BIO read and write callbacks.

**Parameters:**

*ssl* SSL context  
*f\_recv* read callback  
*p\_recv* read parameter  
*f\_send* write callback  
*p\_send* write parameter

Definition at line 1610 of file `ssl_tls.c`.

References `_ssl_context::f_recv`, `_ssl_context::f_send`, `_ssl_context::p_recv`, and `_ssl_context::p_send`.

Referenced by `main()`, and `ssl_test()`.

**13.135.1.24** `void ssl_set_ca_chain (ssl_context * ssl, x509_cert * ca_chain, char * peer_cn)`

Set the data required to verify peer certificate.

**Parameters:**

*ssl* SSL context  
*ca\_chain* trusted CA chain  
*peer\_cn* expected peer CommonName (or NULL)

**Note:**

TODO: add two more parameters: depth and `crl`

Definition at line 1641 of file `ssl_tls.c`.

References `_ssl_context::ca_chain`, and `_ssl_context::peer_cn`.

Referenced by `main()`, and `ssl_test()`.

**13.135.1.25** `void ssl_set_ciphers (ssl_context * ssl, int * ciphers)`

Set the list of allowed ciphersuites.

**Parameters:**

*ssl* SSL context  
*ciphers* 0-terminated list of allowed ciphers

Definition at line 1636 of file `ssl_tls.c`.

References `_ssl_context::ciphers`.

Referenced by `main()`, and `ssl_test()`.

**13.135.1.26** `void ssl_set_dbg (ssl_context * ssl, void(*)(void *, int, char *) f_dbg, void * p_dbg)`

Set the debug callback.

**Parameters:**

*ssl* SSL context

*f\_dbg* debug function

*p\_dbg* debug parameter

Definition at line 1602 of file ssl\_tls.c.

References \_ssl\_context::f\_dbg, and \_ssl\_context::p\_dbg.

Referenced by main(), and ssl\_test().

**13.135.1.27** `int ssl_set_dh_param (ssl_context * ssl, char * dhm_P, char * dhm_G)`

Set the Diffie-Hellman public P and G values, read as hexadecimal strings (server-side only).

**Parameters:**

*ssl* SSL context

*dhm\_P* Diffie-Hellman-Merkle modulus

*dhm\_G* Diffie-Hellman-Merkle generator

**Returns:**

0 if successful

Definition at line 1655 of file ssl\_tls.c.

References \_ssl\_context::dhm\_ctx, dhm\_context::G, mpi\_read\_string(), dhm\_context::P, and SSL\_DEBUG\_RET.

Referenced by main(), and ssl\_test().

**13.135.1.28** `void ssl_set_endpoint (ssl_context * ssl, int endpoint)`

Set the current endpoint type.

**Parameters:**

*ssl* SSL context

*endpoint* must be SSL\_IS\_CLIENT or SSL\_IS\_SERVER

Definition at line 1584 of file ssl\_tls.c.

References \_ssl\_context::endpoint.

Referenced by main(), and ssl\_test().

**13.135.1.29** `int ssl_set_hostname (ssl_context * ssl, char * hostname)`

Set hostname for ServerName TLS Extension.

**Parameters:**

*ssl* SSL context  
*hostname* the server hostname

**Returns:**

0 if successful

Definition at line 1674 of file ssl\_tls.c.

References `_ssl_context::hostname`, `_ssl_context::hostname_len`, and `XYSSL_ERR_SSL_BAD_INPUT_DATA`.

Referenced by `main()`.

**13.135.1.30 void ssl\_set\_own\_cert (ssl\_context \* ssl, x509\_cert \* own\_cert, rsa\_context \* rsa\_key)**

Set own certificate and private key.

**Parameters:**

*ssl* SSL context  
*own\_cert* own public certificate  
*rsa\_key* own private RSA key

Definition at line 1648 of file ssl\_tls.c.

References `_ssl_context::own_cert`, and `_ssl_context::rsa_key`.

Referenced by `main()`, and `ssl_test()`.

**13.135.1.31 void ssl\_set\_rng (ssl\_context \* ssl, int(\*)(void \*)f\_rng, void \* p\_rng)**

Set the random number generator callback.

**Parameters:**

*ssl* SSL context  
*f\_rng* RNG function  
*p\_rng* RNG parameter

Definition at line 1594 of file ssl\_tls.c.

References `_ssl_context::f_rng`, and `_ssl_context::p_rng`.

Referenced by `main()`, and `ssl_test()`.

**13.135.1.32 void ssl\_set\_scb (ssl\_context \* ssl, int(\*)(ssl\_context \*) s\_get, int(\*)(ssl\_context \*) s\_set)**

Set the session callbacks (server-side only).

**Parameters:**

*ssl* SSL context



*s\_get* session get callback

*s\_set* session set callback

Definition at line 1620 of file ssl\_tls.c.

References `_ssl_context::s_get`, and `_ssl_context::s_set`.

Referenced by `main()`.

#### 13.135.1.33 `void ssl_set_session(ssl_context *ssl, int resume, int timeout, ssl_session *session)`

Set the session resuming flag, timeout and data.

##### Parameters:

*ssl* SSL context

*resume* if 0 (default), the session will not be resumed

*timeout* session timeout in seconds, or 0 (no timeout)

*session* session context

Definition at line 1628 of file ssl\_tls.c.

References `_ssl_context::resume`, `_ssl_context::session`, and `_ssl_context::timeout`.

Referenced by `main()`, and `ssl_test()`.

#### 13.135.1.34 `int ssl_write(ssl_context *ssl, unsigned char *buf, int len)`

Write exactly 'len' application data bytes.

##### Parameters:

*ssl* SSL context

*buf* buffer holding the data

*len* how many bytes must be written

##### Returns:

This function returns the number of bytes written, or a negative error code.

##### Note:

When this function returns `XYSSL_ERR_NET_TRY_AGAIN`, it must be called later with the *\*same\** arguments, until it returns a positive value.

Definition at line 1857 of file ssl\_tls.c.

References `_ssl_context::out_left`, `_ssl_context::out_msg`, `_ssl_context::out_msglen`, `_ssl_context::out_msgtype`, `SSL_DEBUG_MSG`, `SSL_DEBUG_RET`, `ssl_flush_output()`, `ssl_handshake()`, `SSL_HANDSHAKE_OVER`, `SSL_MAX_CONTENT_LEN`, `SSL_MSG_APPLICATION_DATA`, `ssl_write_record()`, and `_ssl_context::state`.

Referenced by `main()`, and `ssl_test()`.

**13.135.1.35 int ssl\_write\_certificate (ssl\_context \* ssl)**

Definition at line 1044 of file ssl\_tls.c.

References `_ssl_context::client_auth`, `_ssl_context::endpoint`, `_x509_buf::len`, `_ssl_context::minor_ver`, `_x509_cert::next`, `_ssl_context::out_msg`, `_ssl_context::out_msglen`, `_ssl_context::out_msgtype`, `_ssl_context::own_cert`, `_x509_buf::p`, `_x509_cert::raw`, `SSL_ALERT_NO_CERTIFICATE`, `SSL_ALERT_WARNING`, `SSL_DEBUG_CRT`, `SSL_DEBUG_MSG`, `SSL_DEBUG_RET`, `SSL_HS_CERTIFICATE`, `SSL_IS_CLIENT`, `SSL_MAX_CONTENT_LEN`, `SSL_MINOR_VERSION_0`, `SSL_MSG_ALERT`, `SSL_MSG_HANDSHAKE`, `ssl_write_record()`, `_ssl_context::state`, `XYSSL_ERR_SSL_CERTIFICATE_REQUIRED`, and `XYSSL_ERR_SSL_CERTIFICATE_TOO_LARGE`.

Referenced by `ssl_handshake_client()`, and `ssl_handshake_server()`.

**13.135.1.36 int ssl\_write\_change\_cipher\_spec (ssl\_context \* ssl)**

Definition at line 1287 of file ssl\_tls.c.

References `_ssl_context::do_crypt`, `_ssl_context::out_msg`, `_ssl_context::out_msglen`, `_ssl_context::out_msgtype`, `SSL_DEBUG_MSG`, `SSL_DEBUG_RET`, `SSL_MSG_CHANGE_CIPHER_SPEC`, `ssl_write_record()`, and `_ssl_context::state`.

Referenced by `ssl_handshake_client()`, and `ssl_handshake_server()`.

**13.135.1.37 int ssl\_write\_finished (ssl\_context \* ssl)**

Definition at line 1433 of file ssl\_tls.c.

References `_ssl_context::do_crypt`, `_ssl_context::endpoint`, `_ssl_context::fin_md5`, `_ssl_context::fin_sha1`, `md5()`, `_ssl_context::minor_ver`, `_ssl_context::out_msg`, `_ssl_context::out_msglen`, `_ssl_context::out_msgtype`, `_ssl_context::resume`, `sha1()`, `ssl_calc_finished()`, `SSL_CLIENT_CHANGE_CIPHER_SPEC`, `SSL_DEBUG_MSG`, `SSL_DEBUG_RET`, `SSL_HANDSHAKE_OVER`, `SSL_HS_FINISHED`, `SSL_IS_CLIENT`, `SSL_MINOR_VERSION_0`, `SSL_MSG_HANDSHAKE`, `ssl_write_record()`, and `_ssl_context::state`.

Referenced by `ssl_handshake_client()`, and `ssl_handshake_server()`.

**13.135.1.38 int ssl\_write\_record (ssl\_context \* ssl)**

Definition at line 786 of file ssl\_tls.c.

References `_ssl_context::do_crypt`, `_ssl_context::fin_md5`, `_ssl_context::fin_sha1`, `_ssl_context::major_ver`, `md5_update()`, `_ssl_context::minor_ver`, `_ssl_context::out_hdr`, `_ssl_context::out_left`, `_ssl_context::out_msg`, `_ssl_context::out_msglen`, `_ssl_context::out_msgtype`, `sha1_update()`, `SSL_DEBUG_BUF`, `SSL_DEBUG_MSG`, `SSL_DEBUG_RET`, `ssl_encrypt_buf()`, `ssl_flush_output()`, and `SSL_MSG_HANDSHAKE`.

Referenced by `ssl_close_notify()`, `ssl_write()`, `ssl_write_certificate()`, `ssl_write_certificate_request()`, `ssl_write_certificate_verify()`, `ssl_write_change_cipher_spec()`, `ssl_write_client_hello()`, `ssl_write_client_key_exchange()`, `ssl_write_finished()`, `ssl_write_server_hello()`, `ssl_write_server_hello_done()`, and `ssl_write_server_key_exchange()`.

**13.135.1.39** `static int tls1_prf (unsigned char * secret, int slen, char * label, unsigned char *  
random, int rlen, unsigned char * dstbuf, int dlen) [static]`

Definition at line 46 of file ssl\_tls.c.

References md5\_hmac(), S1, S2, sha1\_hmac(), and XYSSL\_ERR\_SSL\_BAD\_INPUT\_DATA.

Referenced by ssl\_calc\_finished(), and ssl\_derive\_keys().

## 13.135.2 Variable Documentation

**13.135.2.1** `int ssl_default_ciphers[ ]`

**Initial value:**

```
{  
  
    SSL_EDH_RSA_AES_256_SHA,  
  
    SSL_EDH_RSA_DES_168_SHA,  
  
    SSL_RSA_AES_128_SHA,  
    SSL_RSA_AES_256_SHA,  
  
    SSL_RSA_DES_168_SHA,  
  
    SSL_RSA_RC4_128_SHA,  
    SSL_RSA_RC4_128_MD5,  
  
    0  
}
```

Definition at line 1739 of file ssl\_tls.c.

Referenced by main(), and ssl\_test().

## 13.136 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/library/timing.c File Reference

```
#include "xyssl/config.h"
#include "xyssl/timing.h"
#include <unistd.h>
#include <sys/types.h>
#include <sys/time.h>
#include <signal.h>
#include <time.h>
```

### Data Structures

- struct [\\_hr\\_time](#)

### Functions

- unsigned long [hardclock](#) (void)  
*Return the CPU cycle counter value.*
- unsigned long [get\\_timer](#) (struct [hr\\_time](#) \*val, int reset)  
*Return the elapsed time in milliseconds.*
- static void [sighandler](#) (int signum)
- void [set\\_alarm](#) (int seconds)  
*Setup an alarm clock.*
- void [m\\_sleep](#) (int milliseconds)  
*Sleep for a certain amount of time.*

### Variables

- static int [hardclock\\_init](#) = 0
- static struct timeval [tv\\_init](#)
- int [alarmed](#) = 0

#### 13.136.1 Function Documentation

##### 13.136.1.1 unsigned long get\_timer (struct hr\_time \* val, int reset)

Return the elapsed time in milliseconds.

**Parameters:**

*val* points to a timer structure

*reset* if set to 1, the timer is restarted

Definition at line 205 of file timing.c.

References `_hr_time::start`.

Referenced by `ssl_test()`.

**13.136.1.2 unsigned long hardclock (void)**

Return the CPU cycle counter value.

Definition at line 136 of file timing.c.

References `tv_init`.

Referenced by `main()`.

**13.136.1.3 void m\_sleep (int milliseconds)**

Sleep for a certain amount of time.

Definition at line 238 of file timing.c.

**13.136.1.4 void set\_alarm (int seconds)**

Setup an alarm clock.

**Parameters:**

*seconds* delay before the "alarmed" flag is set

Definition at line 231 of file timing.c.

References `sighandler()`.

Referenced by `main()`.

**13.136.1.5 static void sighandler (int signum) [static]**

Definition at line 225 of file timing.c.

Referenced by `set_alarm()`.

**13.136.2 Variable Documentation****13.136.2.1 int alarmed = 0**

Definition at line 159 of file timing.c.

Referenced by `main()`.

**13.136.2.2   int hardclock\_init = 0   [static]**

Definition at line 133 of file timing.c.

**13.136.2.3   struct timeval tv\_init   [static]**

Definition at line 134 of file timing.c.

Referenced by hardclock().

## 13.137 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/library/x509parse.c File Reference

```
#include "xyssl/config.h"
#include "xyssl/x509.h"
#include "xyssl/base64.h"
#include "xyssl/des.h"
#include "xyssl/md2.h"
#include "xyssl/md4.h"
#include "xyssl/md5.h"
#include "xyssl/sha1.h"
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <time.h>
#include "xyssl/certs.h"
```

### Functions

- static [int asn1\\_get\\_len](#) (unsigned char \*\*p, unsigned char \*end, [int](#) \*len)
- static [int asn1\\_get\\_tag](#) (unsigned char \*\*p, unsigned char \*end, [int](#) \*len, [int](#) tag)
- static [int asn1\\_get\\_bool](#) (unsigned char \*\*p, unsigned char \*end, [int](#) \*val)
- static [int asn1\\_get\\_int](#) (unsigned char \*\*p, unsigned char \*end, [int](#) \*val)
- static [int asn1\\_get\\_mpi](#) (unsigned char \*\*p, unsigned char \*end, [mpi](#) \*X)
- static [int x509\\_get\\_version](#) (unsigned char \*\*p, unsigned char \*end, [int](#) \*ver)
- static [int x509\\_get\\_serial](#) (unsigned char \*\*p, unsigned char \*end, [x509\\_buf](#) \*serial)
- static [int x509\\_get\\_alg](#) (unsigned char \*\*p, unsigned char \*end, [x509\\_buf](#) \*alg)
- static [int x509\\_get\\_name](#) (unsigned char \*\*p, unsigned char \*end, [x509\\_name](#) \*cur)
- static [int x509\\_get\\_dates](#) (unsigned char \*\*p, unsigned char \*end, [x509\\_time](#) \*from, [x509\\_time](#) \*to)
- static [int x509\\_get\\_pubkey](#) (unsigned char \*\*p, unsigned char \*end, [x509\\_buf](#) \*pk\_alg\_oid, [mpi](#) \*N, [mpi](#) \*E)
- static [int x509\\_get\\_sig](#) (unsigned char \*\*p, unsigned char \*end, [x509\\_buf](#) \*sig)
- static [int x509\\_get\\_uid](#) (unsigned char \*\*p, unsigned char \*end, [x509\\_buf](#) \*uid, [int](#) n)
- static [int x509\\_get\\_ext](#) (unsigned char \*\*p, unsigned char \*end, [x509\\_buf](#) \*ext, [int](#) \*ca\_istrue, [int](#) \*max\_pathlen)
- [int x509parse\\_crt](#) ([x509\\_cert](#) \*chain, unsigned char \*buf, [int](#) buflen)  
*Parse one or more certificates and add them to the chained list.*
- [int x509parse\\_crtfile](#) ([x509\\_cert](#) \*chain, char \*path)  
*Load one or more certificates and add them to the chained list.*
- static [int x509\\_get\\_iv](#) (unsigned char \*s, unsigned char iv[8])
- static void [x509\\_des3\\_decrypt](#) (unsigned char des3\_iv[8], unsigned char \*buf, [int](#) buflen, unsigned char \*pwd, [int](#) pwdden)

- `int x509parse_key (rsa_context *rsa, unsigned char *buf, int buflen, unsigned char *pwd, int pwrlen)`  
*Parse a private RSA key.*
- `int x509parse_keyfile (rsa_context *rsa, char *path, char *pwd)`  
*Load and parse a private RSA key.*
- `int x509parse_dn_gets (char *buf, char *end, x509_name *dn)`  
*Store the certificate DN in printable form into buf; no more than (end - buf) characters will be written.*
- `char * x509parse_cert_info (char *prefix, x509_cert *crt)`  
*Returns an informational string about the certificate.*
- `int x509parse_expired (x509_cert *crt)`  
*Return 0 if the certificate is still valid, or BADCERT\_EXPIRED.*
- `static void x509_hash (unsigned char *in, int len, int alg, unsigned char *out)`
- `int x509parse_verify (x509_cert *crt, x509_cert *trust_ca, char *cn, int *flags)`  
*Verify the certificate signature.*
- `void x509_free (x509_cert *crt)`  
*Unallocate all certificate data.*
- `int x509_self_test (int verbose)`  
*Checkup routine.*

### 13.137.1 Function Documentation

#### 13.137.1.1 `static int asn1_get_bool (unsigned char **p, unsigned char *end, int *val) [static]`

Definition at line 108 of file x509parse.c.

References ASN1\_BOOLEAN, `asn1_get_tag()`, and XYSSL\_ERR\_ASN1\_INVALID\_LENGTH.

Referenced by `x509_get_ext()`.

#### 13.137.1.2 `static int asn1_get_int (unsigned char **p, unsigned char *end, int *val) [static]`

Definition at line 126 of file x509parse.c.

References `asn1_get_tag()`, ASN1\_INTEGER, and XYSSL\_ERR\_ASN1\_INVALID\_LENGTH.

Referenced by `x509_get_ext()`, `x509_get_version()`, and `x509parse_key()`.

#### 13.137.1.3 `static int asn1_get_len (unsigned char **p, unsigned char *end, int *len) [static]`

Definition at line 52 of file x509parse.c.

References XYSSL\_ERR\_ASN1\_INVALID\_LENGTH, and XYSSL\_ERR\_ASN1\_OUT\_OF\_DATA.

Referenced by `asn1_get_tag()`, `x509_get_name()`, and `x509_get_serial()`.



**13.137.1.4 static int asn1\_get\_mpi (unsigned char \*\**p*, unsigned char \**end*, mpi \**X*)  
[static]**

Definition at line 149 of file x509parse.c.

References `asn1_get_tag()`, `ASN1_INTEGER`, and `mpi_read_binary()`.

Referenced by `x509_get_pubkey()`, and `x509parse_key()`.

**13.137.1.5 static int asn1\_get\_tag (unsigned char \*\**p*, unsigned char \**end*, int \**len*, int *tag*)  
[static]**

Definition at line 93 of file x509parse.c.

References `asn1_get_len()`, `XYSSL_ERR_ASN1_OUT_OF_DATA`, and `XYSSL_ERR_ASN1_UNEXPECTED_TAG`.

Referenced by `asn1_get_bool()`, `asn1_get_int()`, `asn1_get_mpi()`, `x509_get_alg()`, `x509_get_dates()`, `x509_get_ext()`, `x509_get_name()`, `x509_get_pubkey()`, `x509_get_sig()`, `x509_get_uid()`, `x509_get_version()`, `x509parse_crt()`, and `x509parse_key()`.

**13.137.1.6 static void x509\_des3\_decrypt (unsigned char *des3\_iv*[8], unsigned char \**buf*, int  
*buflen*, unsigned char \**pwd*, int *pwdlen*) [static]**

Definition at line 1042 of file x509parse.c.

References `des3_crypt_cbc()`, `des3_set3key_dec()`, `DES_DECRYPT`, `md5_finish()`, `md5_starts()`, and `md5_update()`.

Referenced by `x509parse_key()`.

**13.137.1.7 void x509\_free (x509\_cert \**crt*)**

Unallocate all certificate data.

Definition at line 1613 of file x509parse.c.

References `_x509_cert::issuer`, `_x509_buf::len`, `_x509_cert::next`, `_x509_name::next`, `_x509_buf::p`, `_x509_cert::raw`, `_x509_cert::rsa`, `rsa_free()`, and `_x509_cert::subject`.

Referenced by `main()`, `ssl_free()`, `ssl_test()`, `x509_self_test()`, and `x509parse_crt()`.

**13.137.1.8 static int x509\_get\_alg (unsigned char \*\**p*, unsigned char \**end*, x509\_buf \**alg*)  
[static]**

Definition at line 229 of file x509parse.c.

References `ASN1_CONSTRUCTED`, `asn1_get_tag()`, `ASN1_NULL`, `ASN1_OID`, `ASN1_SEQUENCE`, `_x509_buf::len`, `_x509_buf::p`, `_x509_buf::tag`, `XYSSL_ERR_ASN1_LENGTH_MISMATCH`, and `XYSSL_ERR_X509_CERT_INVALID_ALG`.

Referenced by `x509_get_pubkey()`, and `x509parse_crt()`.

**13.137.1.9** `static int x509_get_dates (unsigned char **p, unsigned char *end, x509_time *from, x509_time *to) [static]`

Definition at line 358 of file x509parse.c.

References ASN1\_CONSTRUCTED, asn1\_get\_tag(), ASN1\_SEQUENCE, ASN1\_UTC\_TIME, \_x509\_time::day, \_x509\_time::hour, \_x509\_time::min, \_x509\_time::mon, \_x509\_time::sec, XYSSL\_ERR\_ASN1\_LENGTH\_MISMATCH, XYSSL\_ERR\_X509\_CERT\_INVALID\_DATE, and \_x509\_time::year.

Referenced by x509parse\_crt().

**13.137.1.10** `static int x509_get_ext (unsigned char **p, unsigned char *end, x509_buf *ext, int *ca_istrue, int *max_pathlen) [static]`

Definition at line 530 of file x509parse.c.

References ASN1\_CONSTRUCTED, ASN1\_CONTEXT\_SPECIFIC, asn1\_get\_bool(), asn1\_get\_int(), asn1\_get\_tag(), ASN1\_OCTET\_STRING, ASN1\_SEQUENCE, \_x509\_buf::len, \_x509\_buf::p, \_x509\_buf::tag, XYSSL\_ERR\_ASN1\_LENGTH\_MISMATCH, XYSSL\_ERR\_ASN1\_UNEXPECTED\_TAG, and XYSSL\_ERR\_X509\_CERT\_INVALID\_EXTENSIONS.

Referenced by x509parse\_crt().

**13.137.1.11** `static int x509_get_iv (unsigned char *s, unsigned char iv[8]) [static]`

Definition at line 1018 of file x509parse.c.

References XYSSL\_ERR\_X509\_KEY\_INVALID\_ENC\_IV.

Referenced by x509parse\_key().

**13.137.1.12** `static int x509_get_name (unsigned char **p, unsigned char *end, x509_name *cur) [static]`

Definition at line 276 of file x509parse.c.

References ASN1\_BMP\_STRING, ASN1\_CONSTRUCTED, asn1\_get\_len(), asn1\_get\_tag(), ASN1\_IA5\_STRING, ASN1\_OID, ASN1\_PRINTABLE\_STRING, ASN1\_SEQUENCE, ASN1\_SET, ASN1\_T61\_STRING, ASN1\_UNIVERSAL\_STRING, ASN1\_UTF8\_STRING, \_x509\_buf::len, \_x509\_name::next, \_x509\_name::oid, \_x509\_buf::p, \_x509\_buf::tag, \_x509\_name::val, XYSSL\_ERR\_ASN1\_LENGTH\_MISMATCH, XYSSL\_ERR\_ASN1\_OUT\_OF\_DATA, XYSSL\_ERR\_ASN1\_UNEXPECTED\_TAG, and XYSSL\_ERR\_X509\_CERT\_INVALID\_NAME.

Referenced by x509parse\_crt().

**13.137.1.13** `static int x509_get_pubkey (unsigned char **p, unsigned char *end, x509_buf *pk_alg_oid, mpi *N, mpi *E) [static]`

Definition at line 421 of file x509parse.c.

References ASN1\_BIT\_STRING, ASN1\_CONSTRUCTED, asn1\_get\_mpi(), asn1\_get\_tag(), ASN1\_SEQUENCE, \_x509\_buf::len, OID\_PKCS1\_RSA, \_x509\_buf::p, x509\_get\_alg(), XYSSL\_ERR\_ASN1\_LENGTH\_MISMATCH, XYSSL\_ERR\_ASN1\_OUT\_OF\_DATA, XYSSL\_ERR\_X509\_CERT\_INVALID\_PUBKEY, and XYSSL\_ERR\_X509\_CERT\_UNKNOWN\_PK\_ALG.

Referenced by x509parse\_crt().

**13.137.1.14** `static int x509_get_serial (unsigned char **p, unsigned char *end, x509_buf *serial)`  
`[static]`

Definition at line 198 of file x509parse.c.

References ASN1\_CONTEXT\_SPECIFIC, asn1\_get\_len(), ASN1\_INTEGER, ASN1\_PRIMITIVE, \_x509\_buf::len, \_x509\_buf::p, \_x509\_buf::tag, XYSSL\_ERR\_ASN1\_OUT\_OF\_DATA, XYSSL\_ERR\_ASN1\_UNEXPECTED\_TAG, and XYSSL\_ERR\_X509\_CERT\_INVALID\_SERIAL.

Referenced by x509parse\_crt().

**13.137.1.15** `static int x509_get_sig (unsigned char **p, unsigned char *end, x509_buf *sig)`  
`[static]`

Definition at line 476 of file x509parse.c.

References ASN1\_BIT\_STRING, asn1\_get\_tag(), \_x509\_buf::len, \_x509\_buf::p, \_x509\_buf::tag, and XYSSL\_ERR\_X509\_CERT\_INVALID\_SIGNATURE.

Referenced by x509parse\_crt().

**13.137.1.16** `static int x509_get_uid (unsigned char **p, unsigned char *end, x509_buf *uid, int n)`  
`[static]`

Definition at line 501 of file x509parse.c.

References ASN1\_CONSTRUCTED, ASN1\_CONTEXT\_SPECIFIC, asn1\_get\_tag(), \_x509\_buf::len, \_x509\_buf::p, \_x509\_buf::tag, and XYSSL\_ERR\_ASN1\_UNEXPECTED\_TAG.

Referenced by x509parse\_crt().

**13.137.1.17** `static int x509_get_version (unsigned char **p, unsigned char *end, int *ver)`  
`[static]`

Definition at line 168 of file x509parse.c.

References ASN1\_CONSTRUCTED, ASN1\_CONTEXT\_SPECIFIC, asn1\_get\_int(), asn1\_get\_tag(), XYSSL\_ERR\_ASN1\_LENGTH\_MISMATCH, XYSSL\_ERR\_ASN1\_UNEXPECTED\_TAG, and XYSSL\_ERR\_X509\_CERT\_INVALID\_VERSION.

Referenced by x509parse\_crt().

**13.137.1.18** `static void x509_hash (unsigned char *in, int len, int alg, unsigned char *out)`  
`[static]`

Definition at line 1481 of file x509parse.c.

References md2(), md4(), md5(), RSA\_MD2, RSA\_MD4, RSA\_MD5, RSA\_SHA1, and sha1().

Referenced by x509parse\_verify().

**13.137.1.19** `int x509_self_test (int verbose)`

Checkup routine.

**Returns:**

0 if successful, or 1 if the test failed

Definition at line 1675 of file x509parse.c.

References `rsa_free()`, `test_ca_cert`, `test_ca_key`, `test_ca_pwd`, `test_cli_cert`, `x509_free()`, `x509parse_cert()`, `x509parse_key()`, and `x509parse_verify()`.

Referenced by `main()`.

**13.137.1.20 char\* x509parse\_cert\_info (char \* prefix, x509\_cert \* crt)**

Returns an informational string about the certificate.

Definition at line 1399 of file x509parse.c.

References `buf`, `_x509_time::day`, `_x509_time::hour`, `_x509_cert::issuer`, `_x509_buf::len`, `_x509_time::min`, `_x509_time::mon`, `mpi::n`, `rsa_context::N`, `_x509_buf::p`, `_x509_cert::rsa`, `RSA_MD2`, `RSA_MD4`, `RSA_MD5`, `RSA_SHA1`, `_x509_time::sec`, `_x509_cert::serial`, `_x509_cert::sig_oid1`, `_x509_cert::subject`, `_x509_cert::valid_from`, `_x509_cert::valid_to`, `_x509_cert::version`, `x509parse_dn_gets()`, and `_x509_time::year`.

Referenced by `debug_print_cert()`, and `main()`.

**13.137.1.21 int x509parse\_cert (x509\_cert \* crt, unsigned char \* buf, int buflen)**

Parse one or more certificates and add them to the chained list.

**Parameters:**

*chain* points to the start of the chain  
*buf* buffer holding the certificate data  
*buflen* size of the buffer

**Returns:**

0 if successful, or a specific X509 error code

Definition at line 647 of file x509parse.c.

References `ASN1_CONSTRUCTED`, `asn1_get_tag()`, `ASN1_SEQUENCE`, `base64_decode()`, `_x509_cert::ca_istrue`, `rsa_context::E`, `_x509_cert::issuer`, `_x509_cert::issuer_id`, `_x509_cert::issuer_raw`, `rsa_context::len`, `_x509_buf::len`, `_x509_cert::max_pathlen`, `mpi_size()`, `rsa_context::N`, `_x509_cert::next`, `OID_PKCS1`, `_x509_buf::p`, `_x509_cert::pk_oid`, `_x509_cert::raw`, `_x509_cert::rsa`, `rsa_check_pubkey()`, `_x509_cert::serial`, `_x509_cert::sig`, `_x509_cert::sig_oid1`, `_x509_cert::sig_oid2`, `_x509_cert::subject`, `_x509_cert::subject_id`, `_x509_cert::subject_raw`, `_x509_cert::tbs`, `_x509_cert::v3_ext`, `_x509_cert::valid_from`, `_x509_cert::valid_to`, `_x509_cert::version`, `x509_free()`, `x509_get_alg()`, `x509_get_dates()`, `x509_get_ext()`, `x509_get_name()`, `x509_get_pubkey()`, `x509_get_serial()`, `x509_get_sig()`, `x509_get_uid()`, `x509_get_version()`, `x509parse_cert()`, `XYSSL_ERR_ASN1_LENGTH_MISMATCH`, `XYSSL_ERR_BASE64_INVALID_CHARACTER`, `XYSSL_ERR_X509_CERT_INVALID_FORMAT`, `XYSSL_ERR_X509_CERT_INVALID_PEM`, `XYSSL_ERR_X509_CERT_SIG_MISMATCH`, `XYSSL_ERR_X509_CERT_UNKNOWN_SIG_ALG`, and `XYSSL_ERR_X509_CERT_UNKNOWN_VERSION`.

Referenced by `main()`, `ssl_parse_certificate()`, `ssl_test()`, `x509_self_test()`, `x509parse_cert()`, and `x509parse_crtdfile()`.

### 13.137.1.22 `int x509parse_crtfile(x509_cert *crt, char *path)`

Load one or more certificates and add them to the chained list.

#### Parameters:

*chain* points to the start of the chain  
*path* filename to read the certificates from

#### Returns:

0 if successful, or a specific X509 error code

Definition at line 979 of file x509parse.c.

References buf, f, and x509parse\_crt().

### 13.137.1.23 `int x509parse_dn_gets(char *buf, char *end, x509_name *dn)`

Store the certificate DN in printable form into buf; no more than (end - buf) characters will be written.

Definition at line 1316 of file x509parse.c.

References \_x509\_buf::len, \_x509\_name::next, \_x509\_name::oid, OID\_PKCS9, OID\_X520, \_x509\_buf::p, PKCS9\_EMAIL, \_x509\_name::val, X520\_COMMON\_NAME, X520\_COUNTRY, X520\_LOCALITY, X520\_ORG\_UNIT, X520\_ORGANIZATION, and X520\_STATE.

Referenced by x509parse\_cert\_info().

### 13.137.1.24 `int x509parse_expired(x509_cert *crt)`

Return 0 if the certificate is still valid, or BADCERT\_EXPIRED.

Definition at line 1458 of file x509parse.c.

References BADCERT\_EXPIRED, \_x509\_time::day, \_x509\_time::mon, \_x509\_cert::valid\_to, and \_x509\_time::year.

Referenced by x509parse\_verify().

### 13.137.1.25 `int x509parse_key(rsa_context *rsa, unsigned char *buf, int buflen, unsigned char *pwd, int pwrlen)`

Parse a private RSA key.

#### Parameters:

*rsa* RSA context to be initialized  
*buf* input buffer  
*buflen* size of the buffer  
*pwd* password for decryption (optional)  
*pwrlen* size of the password

#### Returns:

0 if successful, or a specific X509 error code

Definition at line 1082 of file x509parse.c.

References ASN1\_CONSTRUCTED, asn1\_get\_int(), asn1\_get\_mpi(), asn1\_get\_tag(), ASN1\_SEQUENCE, base64\_decode(), rsa\_context::D, rsa\_context::DP, rsa\_context::DQ, rsa\_context::E, rsa\_context::len, mpi\_size(), rsa\_context::N, rsa\_context::P, rsa\_context::Q, rsa\_context::QP, rsa\_check\_privkey(), rsa\_free(), rsa\_context::ver, x509\_des3\_decrypt(), x509\_get\_iv(), XYSSL\_ERR\_ASN1\_LENGTH\_MISMATCH, XYSSL\_ERR\_BASE64\_INVALID\_CHARACTER, XYSSL\_ERR\_X509\_FEATURE\_UNAVAILABLE, XYSSL\_ERR\_X509\_KEY\_INVALID\_ENC\_IV, XYSSL\_ERR\_X509\_KEY\_INVALID\_FORMAT, XYSSL\_ERR\_X509\_KEY\_INVALID\_PEM, XYSSL\_ERR\_X509\_KEY\_INVALID\_VERSION, XYSSL\_ERR\_X509\_KEY\_PASSWORD\_MISMATCH, XYSSL\_ERR\_X509\_KEY\_PASSWORD\_REQUIRED, and XYSSL\_ERR\_X509\_KEY\_UNKNOWN\_ENC\_ALG.

Referenced by main(), ssl\_test(), x509\_self\_test(), and x509parse\_keyfile().

### 13.137.1.26 int x509parse\_keyfile (rsa\_context \*rsa, char \*path, char \*password)

Load and parse a private RSA key.

#### Parameters:

- rsa* RSA context to be initialized
- path* filename to read the private key from
- pwd* password to decrypt the file (can be NULL)

#### Returns:

- 0 if successful, or a specific X509 error code

Definition at line 1269 of file x509parse.c.

References buf, f, and x509parse\_key().

### 13.137.1.27 int x509parse\_verify (x509\_cert \*crt, x509\_cert \*trust\_ca, char \*cn, int \*flags)

Verify the certificate signature.

#### Parameters:

- crt* a certificate to be verified
- trust\_ca* the trusted CA chain
- cn* expected Common Name (can be set to NULL if the CN must not be verified)
- flags* result of the verification

#### Returns:

- 0 if successful or XYSSL\_ERR\_X509\_SIG\_VERIFY\_FAILED, in which case \*flags will have one or more of the following values set: BADCERT\_EXPIRED -- BADCERT\_REVOKED -- BADCERT\_CN\_MISMATCH -- BADCERT\_NOT\_TRUSTED

#### Note:

- TODO: add two arguments, depth and crl

Definition at line 1503 of file x509parse.c.

References BADCERT\_CN\_MISMATCH, BADCERT\_NOT\_TRUSTED, \_x509\_cert::ca\_istrue, cur, \_x509\_cert::issuer\_raw, \_x509\_buf::len, \_x509\_cert::max\_pathlen, \_x509\_cert::next, \_x509\_name::next, \_x509\_name::oid, OID\_CN, \_x509\_buf::p, \_x509\_cert::rsa, rsa\_pkcs1\_verify(), RSA\_PUBLIC, \_x509\_cert::sig, \_x509\_cert::sig\_oid1, \_x509\_cert::subject, \_x509\_cert::subject\_raw, \_x509\_cert::tbs, \_x509\_name::val, \_x509\_cert::version, x509\_hash(), x509parse\_expired(), and XYSSL\_ERR\_X509\_CERT\_VERIFY\_FAILED.

Referenced by ssl\_parse\_certificate(), and x509\_self\_test().

## 13.138 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/programs/aes/aescript2.c

### File Reference

```
#include <sys/types.h>
#include <unistd.h>
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <time.h>
#include "xyssl/aes.h"
#include "xyssl/sha2.h"
```

### Defines

- #define [MODE\\_ENCRYPT](#) 0
- #define [MODE\\_DECRYPT](#) 1
- #define [USAGE](#)

### Functions

- [int main](#) ([int](#) argc, [char](#) \*argv[ ])

#### 13.138.1 Define Documentation

##### 13.138.1.1 #define MODE\_DECRYPT 1

Definition at line 42 of file aescript2.c.

##### 13.138.1.2 #define MODE\_ENCRYPT 0

Definition at line 41 of file aescript2.c.

##### 13.138.1.3 #define USAGE

#### Value:

```
"\n  aescript2 <mode> <input filename> <output filename> <key>\n" \
"\n  <mode>: 0 = encrypt, 1 = decrypt\n" \
"\n  example: aescript2 0 file file.aes hex:E76B2413958B00E193\n" \
"\n"
```

Definition at line 44 of file aescript2.c.

Referenced by [main\(\)](#).



13.138

/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/programs/aes/aesencrypt2.c

File Reference

917

---

### 13.138.2 Function Documentation

#### 13.138.2.1 `int main (int argc, char * argv[])`

Definition at line 50 of file aesencrypt2.c.

References `aes_crypt_ecb()`, `AES_DECRYPT`, `AES_ENCRYPT`, `aes_setkey_dec()`, `aes_setkey_enc()`, `int`, `MODE_DECRYPT`, `MODE_ENCRYPT`, `sha2_finish()`, `sha2_hmac_finish()`, `sha2_hmac_starts()`, `sha2_hmac_update()`, `sha2_starts()`, `sha2_update()`, and `USAGE`.

## 13.139 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/programs/hash/hello.c **File Reference**

```
#include <stdio.h>
#include "xyssl/md5.h"
```

### Functions

- [int main](#) (void)

#### 13.139.1 Function Documentation

##### 13.139.1.1 int main (void)

Definition at line 29 of file hello.c.

References [md5\(\)](#).

## 13.140 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/programs/hash/md5sum.c

### File Reference

```
#include <string.h>
#include <stdio.h>
#include "xyssl/md5.h"
```

### Functions

- static [int md5\\_wrapper](#) (char \*filename, unsigned char \*sum)
- static [int md5\\_print](#) (char \*filename)
- static [int md5\\_check](#) (char \*filename)
- [int main](#) (int argc, char \*argv[ ])

### 13.140.1 Function Documentation

#### 13.140.1.1 [int main](#) (int *argc*, char \* *argv*[ ])

Definition at line 131 of file md5sum.c.

References [md5\\_check\(\)](#), and [md5\\_print\(\)](#).

#### 13.140.1.2 [static int md5\\_check](#) (char \* *filename*) [[static](#)]

Definition at line 58 of file md5sum.c.

References [buf](#), [f](#), and [md5\\_wrapper\(\)](#).

Referenced by [main\(\)](#).

#### 13.140.1.3 [static int md5\\_print](#) (char \* *filename*) [[static](#)]

Definition at line 43 of file md5sum.c.

References [md5\\_wrapper\(\)](#).

Referenced by [main\(\)](#).

#### 13.140.1.4 [static int md5\\_wrapper](#) (char \* *filename*, unsigned char \* *sum*) [[static](#)]

Definition at line 30 of file md5sum.c.

References [md5\\_file\(\)](#).

Referenced by [md5\\_check\(\)](#), and [md5\\_print\(\)](#).

## 13.141 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/programs/hash/sha1sum.c

### File Reference

```
#include <string.h>
#include <stdio.h>
#include "xyssl/sha1.h"
```

### Functions

- static [int sha1\\_wrapper](#) (char \*filename, unsigned char \*sum)
- static [int sha1\\_print](#) (char \*filename)
- static [int sha1\\_check](#) (char \*filename)
- [int main](#) (int argc, char \*argv[ ])

#### 13.141.1 Function Documentation

##### 13.141.1.1 [int main](#) (int *argc*, char \* *argv*[ ])

Definition at line 131 of file sha1sum.c.

References [sha1\\_check\(\)](#), and [sha1\\_print\(\)](#).

##### 13.141.1.2 [static int sha1\\_check](#) (char \* *filename*) [[static](#)]

Definition at line 58 of file sha1sum.c.

References [buf](#), [f](#), and [sha1\\_wrapper\(\)](#).

Referenced by [main\(\)](#).

##### 13.141.1.3 [static int sha1\\_print](#) (char \* *filename*) [[static](#)]

Definition at line 43 of file sha1sum.c.

References [sha1\\_wrapper\(\)](#).

Referenced by [main\(\)](#).

##### 13.141.1.4 [static int sha1\\_wrapper](#) (char \* *filename*, unsigned char \* *sum*) [[static](#)]

Definition at line 30 of file sha1sum.c.

References [sha1\\_file\(\)](#).

Referenced by [sha1\\_check\(\)](#), and [sha1\\_print\(\)](#).

## 13.142 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/programs/hash/sha2sum.c

### File Reference

```
#include <string.h>
#include <stdio.h>
#include "xyssl/sha2.h"
```

### Functions

- static [int sha2\\_wrapper](#) (char \*filename, unsigned char \*sum)
- static [int sha2\\_print](#) (char \*filename)
- static [int sha2\\_check](#) (char \*filename)
- [int main](#) (int argc, char \*argv[ ])

#### 13.142.1 Function Documentation

##### 13.142.1.1 [int main](#) (int *argc*, char \* *argv*[ ])

Definition at line 131 of file sha2sum.c.

References [sha2\\_check\(\)](#), and [sha2\\_print\(\)](#).

##### 13.142.1.2 [static int sha2\\_check](#) (char \* *filename*) [[static](#)]

Definition at line 58 of file sha2sum.c.

References [buf](#), [f](#), and [sha2\\_wrapper\(\)](#).

Referenced by [main\(\)](#).

##### 13.142.1.3 [static int sha2\\_print](#) (char \* *filename*) [[static](#)]

Definition at line 43 of file sha2sum.c.

References [sha2\\_wrapper\(\)](#).

Referenced by [main\(\)](#).

##### 13.142.1.4 [static int sha2\\_wrapper](#) (char \* *filename*, unsigned char \* *sum*) [[static](#)]

Definition at line 30 of file sha2sum.c.

References [sha2\\_file\(\)](#).

Referenced by [sha2\\_check\(\)](#), and [sha2\\_print\(\)](#).

## 13.143 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/programs/pkey/dh\_client.c

### File Reference

```
#include <string.h>
#include <stdio.h>
#include "xyssl/net.h"
#include "xyssl/aes.h"
#include "xyssl/dhm.h"
#include "xyssl/rsa.h"
#include "xyssl/sha1.h"
#include "xyssl/havege.h"
```

### Defines

- `#define SERVER_NAME "localhost"`
- `#define SERVER_PORT 11999`

### Functions

- `int main (void)`

#### 13.143.1 Define Documentation

##### 13.143.1.1 `#define SERVER_NAME "localhost"`

Definition at line 35 of file dh\_client.c.

Referenced by main().

##### 13.143.1.2 `#define SERVER_PORT 11999`

Definition at line 36 of file dh\_client.c.

Referenced by main().

#### 13.143.2 Function Documentation

##### 13.143.2.1 `int main (void)`

Definition at line 38 of file dh\_client.c.

References aes\_crypt\_ecb(), AES\_DECRYPT, aes\_setkey\_dec(), buf, dhm\_calc\_secret(), dhm\_free(), dhm\_make\_public(), dhm\_read\_params(), rsa\_context::E, f, havege\_init(), havege\_rand(), dhm\_context::len, rsa\_context::len, mpi\_msb(), mpi\_read\_file(), rsa\_context::N, net\_close(), net\_connect(), net\_recv(), net\_send(), rsa\_free(), rsa\_init(), rsa\_pkcs1\_verify(), RSA\_PKCS\_V15, RSA\_PUBLIC, RSA\_SHA1, server\_fd, SERVER\_NAME, SERVER\_PORT, and sha1().

## **13.144 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/programs/pkey/dh\_genprime.c File Reference**

```
#include <stdio.h>
#include "xyssl/bignum.h"
#include "xyssl/config.h"
#include "xyssl/havege.h"
```

### **Defines**

- `#define DH_P_SIZE 1024`
- `#define GENERATOR "4"`

### **Functions**

- `int main` (void)

#### **13.144.1 Define Documentation**

##### **13.144.1.1 #define DH\_P\_SIZE 1024**

Definition at line 35 of file dh\_genprime.c.

Referenced by main().

##### **13.144.1.2 #define GENERATOR "4"**

Definition at line 36 of file dh\_genprime.c.

Referenced by main().

#### **13.144.2 Function Documentation**

##### **13.144.2.1 int main (void)**

Definition at line 38 of file dh\_genprime.c.

References DH\_P\_SIZE, GENERATOR, havege\_init(), havege\_rand(), mpi\_div\_int(), mpi\_free(), mpi\_gen\_prime(), mpi\_init(), mpi\_is\_prime(), mpi\_read\_string(), mpi\_sub\_int(), mpi\_write\_file(), and P.

## 13.145 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/programs/pkey/dh\_server.c

### File Reference

```
#include <string.h>
#include <stdio.h>
#include "xyssl/net.h"
#include "xyssl/aes.h"
#include "xyssl/dhm.h"
#include "xyssl/rsa.h"
#include "xyssl/sha1.h"
#include "xyssl/havege.h"
```

### Defines

- `#define SERVER_PORT 11999`
- `#define PLAINTEXT "==Hello there!=="`

### Functions

- `int main (void)`

#### 13.145.1 Define Documentation

##### 13.145.1.1 `#define PLAINTEXT "==Hello there!=="`

Definition at line 36 of file dh\_server.c.

Referenced by main().

##### 13.145.1.2 `#define SERVER_PORT 11999`

Definition at line 35 of file dh\_server.c.

#### 13.145.2 Function Documentation

##### 13.145.2.1 `int main (void)`

Definition at line 38 of file dh\_server.c.

References aes\_crypt\_ecb(), AES\_ENCRYPT, aes\_setkey\_enc(), buf, client\_fd, rsa\_context::D, dhm\_calc\_secret(), dhm\_free(), dhm\_make\_params(), dhm\_read\_public(), rsa\_context::DP, rsa\_context::DQ, rsa\_context::E, f, dhm\_context::G, havege\_init(), havege\_rand(), dhm\_context::len, rsa\_context::len, mpi\_msb(), mpi\_read\_file(), rsa\_context::N, net\_accept(), net\_bind(), net\_close(), net\_recv(), net\_send(), dhm\_context::P, rsa\_context::P, PLAINTEXT, rsa\_context::Q, rsa\_context::QP, rsa\_free(), rsa\_init(), rsa\_pkcs1\_sign(), RSA\_PKCS\_V15, RSA\_PRIVATE, RSA\_SHA1, SERVER\_PORT, and sha1().



## **13.146 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/programs/pkey/mpi\_demo.c File Reference**

```
#include <stdio.h>
#include "xyssl/bignum.h"
```

### **Functions**

- [int main](#) (void)

#### **13.146.1 Function Documentation**

##### **13.146.1.1 int main (void)**

Definition at line 29 of file mpi\_demo.c.

References [mpi\\_exp\\_mod\(\)](#), [mpi\\_free\(\)](#), [mpi\\_init\(\)](#), [mpi\\_inv\\_mod\(\)](#), [mpi\\_mul\\_mpi\(\)](#), [mpi\\_read\\_string\(\)](#), [mpi\\_sub\\_int\(\)](#), [mpi\\_write\\_file\(\)](#), and [P](#).

## 13.147 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/programs/pkey/rsa\_genkey.c

### File Reference

```
#include <stdio.h>
#include "xyssl/havege.h"
#include "xyssl/bignum.h"
#include "xyssl/x509.h"
#include "xyssl/rsa.h"
```

### Defines

- `#define KEY_SIZE 1024`
- `#define EXPONENT 65537`

### Functions

- `int main (void)`

#### 13.147.1 Define Documentation

##### 13.147.1.1 `#define EXPONENT 65537`

Definition at line 33 of file `rsa_genkey.c`.

##### 13.147.1.2 `#define KEY_SIZE 1024`

Definition at line 32 of file `rsa_genkey.c`.

#### 13.147.2 Function Documentation

##### 13.147.2.1 `int main (void)`

Definition at line 35 of file `rsa_genkey.c`.

References `rsa_context::D`, `rsa_context::DP`, `rsa_context::DQ`, `rsa_context::E`, `EXPONENT`, `havege_init()`, `havege_rand()`, `KEY_SIZE`, `mpi_write_file()`, `rsa_context::N`, `rsa_context::P`, `rsa_context::Q`, `rsa_context::QP`, `rsa_free()`, `rsa_gen_key()`, `rsa_init()`, and `RSA_PKCS_V15`.

13.148

/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/programs/pkey/rsa\_sign.c

File Reference

927

**13.148** /home/dko/Projects/mobilec/tags/MobileC-

**v1.10.10/src/security/xyssl-0.9/programs/pkey/rsa\_sign.c**

## **File Reference**

```
#include <string.h>
#include <stdio.h>
#include "xyssl/rsa.h"
#include "xyssl/sha1.h"
```

## **Functions**

- [int main](#) ([int](#) argc, char \*argv[ ])

### **13.148.1 Function Documentation**

#### **13.148.1.1 int main (int *argc*, char \* *argv*[ ])**

Definition at line 31 of file rsa\_sign.c.

References [buf](#), [rsa\\_context::D](#), [rsa\\_context::DP](#), [rsa\\_context::DQ](#), [rsa\\_context::E](#), [f](#), [rsa\\_context::len](#), [mpi\\_msb\(\)](#), [mpi\\_read\\_file\(\)](#), [rsa\\_context::N](#), [rsa\\_context::P](#), [rsa\\_context::Q](#), [rsa\\_context::QP](#), [rsa\\_init\(\)](#), [rsa\\_pkcs1\\_sign\(\)](#), [RSA\\_PKCS\\_V15](#), [RSA\\_PRIVATE](#), [RSA\\_SHA1](#), and [sha1\\_file\(\)](#).

## 13.149 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/programs/pkey/rsa\_verify.c

### File Reference

```
#include <string.h>
#include <stdio.h>
#include "xyssl/rsa.h"
#include "xyssl/sha1.h"
```

### Functions

- `int main (int argc, char *argv[ ])`

#### 13.149.1 Function Documentation

##### 13.149.1.1 `int main (int argc, char * argv[ ])`

Definition at line 31 of file `rsa_verify.c`.

References `buf`, `rsa_context::E`, `f`, `rsa_context::len`, `mpi_msb()`, `mpi_read_file()`, `rsa_context::N`, `rsa_init()`, `rsa_pkcs1_verify()`, `RSA_PKCS_V15`, `RSA_PUBLIC`, `RSA_SHA1`, and `sha1_file()`.

## 13.150 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/programs/ssl/ssl\_client1.c

### File Reference

```
#include <string.h>
#include <stdio.h>
#include "xyssl/net.h"
#include "xyssl/ssl.h"
#include "xyssl/havege.h"
```

### Defines

- `#define SERVER_PORT 443`
- `#define SERVER_NAME "xyssl.org"`
- `#define GET_REQUEST`
- `#define DEBUG_LEVEL 0`

### Functions

- `void my_debug (void *ctx, int level, char *str)`
- `int main (void)`

#### 13.150.1 Define Documentation

##### 13.150.1.1 `#define DEBUG_LEVEL 0`

Definition at line 42 of file `ssl_client1.c`.

Referenced by `my_debug()`.

##### 13.150.1.2 `#define GET_REQUEST`

#### Value:

```
"GET /hello/ HTTP/1.1\r\n" \
  "Host: xyssl.org\r\n\r\n"
```

Definition at line 38 of file `ssl_client1.c`.

Referenced by `main()`.

##### 13.150.1.3 `#define SERVER_NAME "xyssl.org"`

Definition at line 37 of file `ssl_client1.c`.

##### 13.150.1.4 `#define SERVER_PORT 443`

Definition at line 32 of file `ssl_client1.c`.

## 13.150.2 Function Documentation

### 13.150.2.1 `int main (void)`

Definition at line 53 of file `ssl_client1.c`.

References `buf`, `GET_REQUEST`, `havege_init()`, `havege_rand()`, `my_debug()`, `net_close()`, `net_connect()`, `net_recv()`, `net_send()`, `server_fd`, `SERVER_NAME`, `SERVER_PORT`, `ssl_close_notify()`, `ssl_default_ciphers`, `ssl_free()`, `ssl_init()`, `SSL_IS_CLIENT`, `ssl_read()`, `ssl_set_authmode()`, `ssl_set_bio()`, `ssl_set_ciphers()`, `ssl_set_dbg()`, `ssl_set_endpoint()`, `ssl_set_rng()`, `ssl_set_session()`, `SSL_VERIFY_NONE`, `ssl_write()`, `XYSSL_ERR_NET_TRY_AGAIN`, and `XYSSL_ERR_SSL_PEER_CLOSE_NOTIFY`.

### 13.150.2.2 `void my_debug (void * ctx, int level, char * str)`

Definition at line 44 of file `ssl_client1.c`.

References `DEBUG_LEVEL`.

Referenced by `main()`, and `ssl_test()`.

## 13.151 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/programs/ssl/ssl\_client2.c

### File Reference

```
#include <string.h>
#include <stdio.h>
#include "xyssl/net.h"
#include "xyssl/ssl.h"
#include "xyssl/havege.h"
#include "xyssl/certs.h"
#include "xyssl/x509.h"
```

### Defines

- #define [SERVER\\_PORT](#) 443
- #define [SERVER\\_NAME](#) "xyssl.org"
- #define [GET\\_REQUEST](#)
- #define [DEBUG\\_LEVEL](#) 0

### Functions

- void [my\\_debug](#) (void \*ctx, [int](#) level, char \*str)
- [int](#) [main](#) (void)

#### 13.151.1 Define Documentation

##### 13.151.1.1 #define [DEBUG\\_LEVEL](#) 0

Definition at line 44 of file ssl\_client2.c.

##### 13.151.1.2 #define [GET\\_REQUEST](#)

#### Value:

```
"GET /hello/ HTTP/1.1\r\n" \
  "Host: xyssl.org\r\n\r\n"
```

Definition at line 40 of file ssl\_client2.c.

##### 13.151.1.3 #define [SERVER\\_NAME](#) "xyssl.org"

Definition at line 39 of file ssl\_client2.c.

##### 13.151.1.4 #define [SERVER\\_PORT](#) 443

Definition at line 34 of file ssl\_client2.c.

## 13.151.2 Function Documentation

### 13.151.2.1 `int main (void)`

Definition at line 55 of file `ssl_client2.c`.

References `BADCERT_CN_MISMATCH`, `BADCERT_EXPIRED`, `BADCERT_NOT_TRUSTED`, `BADCERT_REVOKED`, `buf`, `GET_REQUEST`, `havege_init()`, `havege_rand()`, `net_close()`, `net_connect()`, `net_recv()`, `net_send()`, `_ssl_context::peer_cert`, `rsa_free()`, `server_fd`, `SERVER_NAME`, `SERVER_PORT`, `ssl_close_notify()`, `ssl_default_ciphers`, `ssl_free()`, `ssl_get_cipher()`, `ssl_get_verify_result()`, `ssl_handshake()`, `ssl_init()`, `SSL_IS_CLIENT`, `ssl_read()`, `ssl_set_authmode()`, `ssl_set_bio()`, `ssl_set_ca_chain()`, `ssl_set_ciphers()`, `ssl_set_endpoint()`, `ssl_set_hostname()`, `ssl_set_own_cert()`, `ssl_set_rng()`, `ssl_set_session()`, `SSL_VERIFY_OPTIONAL`, `ssl_write()`, `test_cli_cert`, `test_cli_key`, `x509_free()`, `x509parse_cert_info()`, `x509parse_cert()`, `x509parse_key()`, `xyssl_ca_cert`, `XYSSL_ERR_NET_TRY_AGAIN`, and `XYSSL_ERR_SSL_PEER_CLOSE_NOTIFY`.

### 13.151.2.2 `void my_debug (void * ctx, int level, char * str)`

Definition at line 46 of file `ssl_client2.c`.

References `DEBUG_LEVEL`.



13.152

/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/programs/ssl/ssl\_server.c

File Reference

933

## 13.152 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/programs/ssl/ssl\_server.c File Reference

```
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include "xyssl/havege.h"
#include "xyssl/certs.h"
#include "xyssl/x509.h"
#include "xyssl/ssl.h"
#include "xyssl/net.h"
```

### Defines

- `#define HTTP_RESPONSE`
- `#define DEBUG_LEVEL 0`

### Functions

- `void my_debug (void *ctx, int level, char *str)`
- `static int my_get_session (ssl_context *ssl)`
- `static int my_set_session (ssl_context *ssl)`
- `int main (void)`

### Variables

- `char * my_dhm_P`
- `char * my_dhm_G = "4"`
- `int my_ciphers []`
- `ssl_session * s_list_1st = NULL`
- `ssl_session * cur`
- `ssl_session * prv`

#### 13.152.1 Define Documentation

##### 13.152.1.1 `#define DEBUG_LEVEL 0`

Definition at line 75 of file `ssl_server.c`.

##### 13.152.1.2 `#define HTTP_RESPONSE`

Value:

```
"HTTP/1.0 200 OK\r\nContent-Type: text/html\r\n\r\n" \
"<h2><p><center>Successful connection using: %s\r\n"
```

Definition at line 39 of file `ssl_server.c`.

Referenced by `http_ParseRequest()`, `main()`, and `mtp_http_Parse()`.

## 13.152.2 Function Documentation

### 13.152.2.1 `int main (void)`

Definition at line 160 of file `ssl_server.c`.

References `buf`, `client_fd`, `havege_init()`, `havege_rand()`, `HTTP_RESPONSE`, `my_ciphers`, `my_debug()`, `my_dhm_G`, `my_dhm_P`, `my_get_session()`, `my_set_session()`, `net_accept()`, `net_bind()`, `net_close()`, `net_recv()`, `net_send()`, `_ssl_session::next`, `x509_cert::next`, `rsa_free()`, `ssl_close_notify()`, `ssl_free()`, `ssl_get_cipher()`, `ssl_handshake()`, `ssl_init()`, `SSL_IS_SERVER`, `ssl_read()`, `ssl_set_authmode()`, `ssl_set_bio()`, `ssl_set_ca_chain()`, `ssl_set_ciphers()`, `ssl_set_dbg()`, `ssl_set_dh_param()`, `ssl_set_endpoint()`, `ssl_set_own_cert()`, `ssl_set_rng()`, `ssl_set_scb()`, `ssl_set_session()`, `SSL_VERIFY_NONE`, `ssl_write()`, `test_ca_cert`, `test_srv_cert`, `test_srv_key`, `x509_free()`, `x509parse_cert()`, `x509parse_key()`, `XYSSL_ERR_NET_CONN_RESET`, `XYSSL_ERR_NET_TRY_AGAIN`, and `XYSSL_ERR_SSL_PEER_CLOSE_NOTIFY`.

### 13.152.2.2 `void my_debug (void * ctx, int level, char * str)`

Definition at line 77 of file `ssl_server.c`.

References `DEBUG_LEVEL`.

### 13.152.2.3 `static int my_get_session (ssl_context * ssl) [static]`

Definition at line 93 of file `ssl_server.c`.

References `_ssl_session::cipher`, `_ssl_session::id`, `_ssl_session::length`, `_ssl_session::master`, `_ssl_session::next`, `_ssl_context::resume`, `_ssl_context::session`, `_ssl_session::start`, and `_ssl_context::timeout`.

Referenced by `main()`.

### 13.152.2.4 `static int my_set_session (ssl_context * ssl) [static]`

Definition at line 125 of file `ssl_server.c`.

References `_ssl_session::id`, `_ssl_session::length`, `_ssl_session::next`, `_ssl_context::session`, `_ssl_session::start`, and `_ssl_context::timeout`.

Referenced by `main()`.

## 13.152.3 Variable Documentation

### 13.152.3.1 `ssl_session* cur`

Definition at line 91 of file `ssl_server.c`.

Referenced by `http_GetToken()`, `http_ParseRequest()`, and `x509parse_verify()`.

**13.152**

**/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/programs/ssl/ssl\_server.c**

**File Reference**

**935**

**13.152.3.2 int my\_ciphers[]**

---

**Initial value:**

```
{
    SSL_EDH_RSA_AES_256_SHA,
    SSL_EDH_RSA_DES_168_SHA,
    SSL_RSA_AES_256_SHA,
    SSL_RSA_AES_128_SHA,
    SSL_RSA_DES_168_SHA,
    SSL_RSA_RC4_128_SHA,
    SSL_RSA_RC4_128_MD5,
    0
}
```

Definition at line 63 of file ssl\_server.c.

Referenced by main().

**13.152.3.3 char\* my\_dhm\_G = "4"**

Definition at line 58 of file ssl\_server.c.

Referenced by main().

**13.152.3.4 char\* my\_dhm\_P**

**Initial value:**

```
"E4004C1F94182000103D883A448B3F80"
"2CE4B44A83301270002C20D0321CFD00"
"11CCEF784C26A400F43DFB901BCA7538"
"F2C6B176001CF5A0FD16D2C48B1D0C1C"
"F6AC8E1DA6BCC3B4E1F96B0564965300"
"FFA1D0B601EB2800F489AA512C4B248C"
"01F76949A60BB7F00A40B1EAB64BDD48"
"E8A700D60B7F1200FA8E77B0A979DABF"
```

Definition at line 48 of file ssl\_server.c.

Referenced by main().

**13.152.3.5 ssl\_session \* prv**

Definition at line 91 of file ssl\_server.c.

Referenced by aes\_self\_test(), and des\_self\_test().

**13.152.3.6 ssl\_session\* s\_list\_1st = NULL**

Definition at line 90 of file ssl\_server.c.

## 13.153 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/programs/test/benchmark.c

### File Reference

```
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include "xyssl/config.h"
#include "xyssl/md4.h"
#include "xyssl/md5.h"
#include "xyssl/sha1.h"
#include "xyssl/sha2.h"
#include "xyssl/arc4.h"
#include "xyssl/des.h"
#include "xyssl/aes.h"
#include "xyssl/rsa.h"
#include "xyssl/timing.h"
```

### Defines

- #define [BUFSIZE](#) 1024

### Functions

- static [int myrand](#) (void \*rng\_state)
- [int main](#) (void)

### Variables

- unsigned char [buf](#) [BUFSIZE]

## 13.153.1 Define Documentation

### 13.153.1.1 #define BUFSIZE 1024

Definition at line 41 of file benchmark.c.

Referenced by [main\(\)](#).

## 13.153.2 Function Documentation

---

### 13.153.2.1 int main (void)

Definition at line 53 of file benchmark.c.

References aes\_crypt\_cbc(), AES\_ENCRYPT, aes\_setkey\_enc(), alarmed, arc4\_crypt(), arc4\_setup(), buf, BUFSIZE, des3\_crypt\_cbc(), des3\_set3key\_enc(), des\_crypt\_cbc(), DES\_ENCRYPT, des\_setkey\_enc(), hardclock(), md4(), md5(), myrand(), rsa\_free(), rsa\_gen\_key(), rsa\_init(), RSA\_PKCS\_V15, rsa\_private(), rsa\_public(), set\_alarm(), sha1(), and sha2().

### 13.153.2.2 static int myrand (void \*rng\_state) [static]

Definition at line 43 of file benchmark.c.

Referenced by main().

## 13.153.3 Variable Documentation

### 13.153.3.1 unsigned char buf[BUFSIZE]

Definition at line 51 of file benchmark.c.

Referenced by aes\_self\_test(), agent\_xml\_compose\_\_create\_row\_nodes(), agent\_xml\_compose\_\_data(), agent\_xml\_compose\_\_task(), agent\_xml\_compose\_\_tasks(), agent\_xml\_parse\_\_fill\_row\_data(), agent\_xml\_parse\_\_tasks(), arc4\_self\_test(), cmd\_prompt\_Thread(), des\_self\_test(), fipa\_DateTime\_Compose(), fipa\_datetime\_Parse(), fipa\_envelope\_Compose\_\_date(), fipa\_envelope\_Compose\_\_from(), fipa\_envelope\_HandleParams(), main(), MC\_Initialize(), MC\_LoadAgentFromFile(), MC\_RetrieveAgentCode(), MC\_SendAgentMigrationMessageFile(), md5\_check(), md5\_file(), md5\_self\_test(), message\_InitializeFromAgent(), message\_send\_Thread(), message\_xml\_parse\_\_message(), mtp\_http\_ComposeMessage(), mtp\_http\_CreateMessage(), mxml\_fd\_getc(), mxml\_fd\_putc(), mxmlLoadFd(), mxmlSaveFd(), net\_htons(), rsa\_pkcs1\_decrypt(), rsa\_pkcs1\_verify(), sha1\_check(), sha1\_file(), sha1\_self\_test(), sha2\_check(), sha2\_file(), sha2\_self\_test(), sha4\_file(), sha4\_self\_test(), ssl\_flush\_output(), ssl\_parse\_client\_hello(), ssl\_parse\_finished(), ssl\_parse\_server\_hello(), ssl\_write\_certificate\_request(), ssl\_write\_client\_hello(), ssl\_write\_server\_hello(), udplisten\_Thread(), x509parse\_cert\_info(), x509parse\_crtfile(), x509parse\_keyfile(), and xml\_get\_cdata().

## 13.154 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/programs/test/selftest.c File Reference

```
#include <string.h>
#include <stdio.h>
#include "xyssl/config.h"
#include "xyssl/md2.h"
#include "xyssl/md4.h"
#include "xyssl/md5.h"
#include "xyssl/sha1.h"
#include "xyssl/sha2.h"
#include "xyssl/sha4.h"
#include "xyssl/arc4.h"
#include "xyssl/des.h"
#include "xyssl/aes.h"
#include "xyssl/base64.h"
#include "xyssl/bignum.h"
#include "xyssl/rsa.h"
#include "xyssl/x509.h"
```

### Functions

- [int main](#) ([int argc](#), [char \\*argv\[ \]](#))

#### 13.154.1 Function Documentation

##### 13.154.1.1 [int main \(int argc, char \\* argv\[ \]\)](#)

Definition at line 44 of file selftest.c.

References [aes\\_self\\_test\(\)](#), [arc4\\_self\\_test\(\)](#), [base64\\_self\\_test\(\)](#), [des\\_self\\_test\(\)](#), [md2\\_self\\_test\(\)](#), [md4\\_self\\_test\(\)](#), [md5\\_self\\_test\(\)](#), [mpi\\_self\\_test\(\)](#), [rsa\\_self\\_test\(\)](#), [sha1\\_self\\_test\(\)](#), [sha2\\_self\\_test\(\)](#), [sha4\\_self\\_test\(\)](#), and [x509\\_self\\_test\(\)](#).

13.155

/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/programs/test/ssl\_test.c

File Reference

939

**13.155** /home/dko/Projects/mobilec/tags/MobileC-

**v1.10.10/src/security/xyssl-0.9/programs/test/ssl\_test.c** **File**

## Reference

```
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include "xyssl/net.h"
#include "xyssl/ssl.h"
#include "xyssl/havege.h"
#include "xyssl/timing.h"
#include "xyssl/certs.h"
```

## Data Structures

- struct [options](#)

## Defines

- #define [OPMODE\\_NONE](#) 0
- #define [OPMODE\\_CLIENT](#) 1
- #define [OPMODE\\_SERVER](#) 2
- #define [IOMODE\\_BLOCK](#) 0
- #define [IOMODE\\_NONBLOCK](#) 1
- #define [COMMAND\\_READ](#) 1
- #define [COMMAND\\_WRITE](#) 2
- #define [COMMAND\\_BOTH](#) 3
- #define [DFL\\_OPMODE](#) OPMODE\_NONE
- #define [DFL\\_IOMODE](#) IOMODE\_BLOCK
- #define [DFL\\_SERVER\\_NAME](#) "localhost"
- #define [DFL\\_SERVER\\_PORT](#) 4433
- #define [DFL\\_COMMAND](#) COMMAND\_READ
- #define [DFL\\_BUFFER\\_SIZE](#) 1024
- #define [DFL\\_MAX\\_BYTES](#) 0
- #define [DFL\\_DEBUG\\_LEVEL](#) 0
- #define [DFL\\_CONN\\_TIMEOUT](#) 0
- #define [DFL\\_MAX\\_CONNECTIONS](#) 0
- #define [DFL\\_SESSION\\_REUSE](#) 1
- #define [DFL\\_SESSION\\_LIFETIME](#) 86400
- #define [DFL\\_FORCE\\_CIPHER](#) 0
- #define [USAGE](#)

## Functions

- unsigned long `int lcppm5` (unsigned long `int *state`)
- void `my_debug` (void `*ctx`, `int level`, char `*str`)
- static `int ssl_test` (struct `options *opt`)
- `int main` (`int argc`, char `*argv[]`)

## Variables

- char `* dhm_G` = "4"
- char `* dhm_P`
- `int server_fd` = -1

### 13.155.1 Define Documentation

#### 13.155.1.1 `#define COMMAND_BOTH 3`

Definition at line 44 of file `ssl_test.c`.

Referenced by `main()`.

#### 13.155.1.2 `#define COMMAND_READ 1`

Definition at line 42 of file `ssl_test.c`.

Referenced by `main()`, and `ssl_test()`.

#### 13.155.1.3 `#define COMMAND_WRITE 2`

Definition at line 43 of file `ssl_test.c`.

Referenced by `main()`, and `ssl_test()`.

#### 13.155.1.4 `#define DFL_BUFFER_SIZE 1024`

Definition at line 51 of file `ssl_test.c`.

Referenced by `main()`.

#### 13.155.1.5 `#define DFL_COMMAND COMMAND_READ`

Definition at line 50 of file `ssl_test.c`.

Referenced by `main()`.

#### 13.155.1.6 `#define DFL_CONN_TIMEOUT 0`

Definition at line 54 of file `ssl_test.c`.

Referenced by `main()`.



### 13.155

/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/programs/test/ssl\_test.c

#### File Reference

941

#### 13.155.1.7 **#define DFL\_DEBUG\_LEVEL 0**

Definition at line 53 of file ssl\_test.c.

Referenced by main().

#### 13.155.1.8 **#define DFL\_FORCE\_CIPHER 0**

Definition at line 58 of file ssl\_test.c.

Referenced by main(), and ssl\_test().

#### 13.155.1.9 **#define DFL\_IOMODE IOMODE\_BLOCK**

Definition at line 47 of file ssl\_test.c.

Referenced by main().

#### 13.155.1.10 **#define DFL\_MAX\_BYTES 0**

Definition at line 52 of file ssl\_test.c.

Referenced by main().

#### 13.155.1.11 **#define DFL\_MAX\_CONNECTIONS 0**

Definition at line 55 of file ssl\_test.c.

Referenced by main().

#### 13.155.1.12 **#define DFL\_OPMODE OPMODE\_NONE**

Definition at line 46 of file ssl\_test.c.

Referenced by main().

#### 13.155.1.13 **#define DFL\_SERVER\_NAME "localhost"**

Definition at line 48 of file ssl\_test.c.

Referenced by main().

#### 13.155.1.14 **#define DFL\_SERVER\_PORT 4433**

Definition at line 49 of file ssl\_test.c.

Referenced by main().

#### 13.155.1.15 **#define DFL\_SESSION\_LIFETIME 86400**

Definition at line 57 of file ssl\_test.c.

Referenced by main().

**13.155.1.16 #define DFL\_SESSION\_REUSE 1**

Definition at line 56 of file ssl\_test.c.

Referenced by main().

**13.155.1.17 #define IOMODE\_BLOCK 0**

Definition at line 39 of file ssl\_test.c.

Referenced by main().

**13.155.1.18 #define IOMODE\_NONBLOCK 1**

Definition at line 40 of file ssl\_test.c.

Referenced by main(), and ssl\_test().

**13.155.1.19 #define OPMODE\_CLIENT 1**

Definition at line 36 of file ssl\_test.c.

Referenced by main(), and ssl\_test().

**13.155.1.20 #define OPMODE\_NONE 0**

Definition at line 35 of file ssl\_test.c.

**13.155.1.21 #define OPMODE\_SERVER 2**

Definition at line 37 of file ssl\_test.c.

Referenced by main(), and ssl\_test().

**13.155.1.22 #define USAGE****Value:**

```
"\n usage: ssl_test opmode=<> command=<>...\n"
"\n acceptable parameters:\n"
"  opmode=client/server      default: <none>\n"
"  iomode=block/nonblock    default: block\n"
"  server_name=%%s          default: localhost\n"
"  server_port=%%d           default: 4433\n"
"  command=read/write/both  default: read\n"
"  buffer_size=%%d (bytes)  default: 1024\n"
"  max_bytes=%%d (bytes)    default: 0 (no limit)\n"
"  debug_level=%%d          default: 0 (disabled)\n"
"  conn_timeout=%%d (ms)    default: 0 (no timeout)\n"
"  max_connections=%%d      default: 0 (no limit)\n"
"  session_reuse=on/off     default: on (enabled)\n"
"  session_lifetime=%%d (s) default: 86400\n"
"  force_cipher=<name>      default: all enabled\n"
" acceptable cipher names:\n"
"  SSL_RSA_RC4_128_MD5      SSL_RSA_RC4_128_SHA\n"
"  SSL_RSA_DES_168_SHA      SSL_EDH_RSA_DES_168_SHA"
```

### 13.155

/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/security/xyssl-0.9/programs/test/ssl\_test.c

#### File Reference

943

```
"    SSL_RSA_AES_128_SHA    SSL_EDH_RSA_AES_256_SHA\n"    SSL_RSA_AES_256_SHA\n"
```

Definition at line 372 of file ssl\_test.c.

## 13.155.2 Function Documentation

### 13.155.2.1 unsigned long int lcppm5 (unsigned long int \* state)

Definition at line 96 of file ssl\_test.c.

Referenced by ssl\_test().

### 13.155.2.2 int main (int argc, char \* argv[])

Definition at line 394 of file ssl\_test.c.

References options::buffer\_size, options::command, COMMAND\_BOTH, COMMAND\_READ, COMMAND\_WRITE, options::conn\_timeout, options::debug\_level, DFL\_BUFFER\_SIZE, DFL\_COMMAND, DFL\_CONN\_TIMEOUT, DFL\_DEBUG\_LEVEL, DFL\_FORCE\_CIPHER, DFL\_IOMODE, DFL\_MAX\_BYTES, DFL\_MAX\_CONNECTIONS, DFL\_OPMODE, DFL\_SERVER\_NAME, DFL\_SERVER\_PORT, DFL\_SESSION\_LIFETIME, DFL\_SESSION\_REUSE, options::force\_cipher, options::iomode, IOMODE\_BLOCK, IOMODE\_NONBLOCK, options::max\_bytes, options::max\_connections, options::opmode, OPMODE\_CLIENT, OPMODE\_SERVER, options::server\_name, options::server\_port, options::session\_lifetime, options::session\_reuse, SSL\_EDH\_RSA\_AES\_256\_SHA, SSL\_EDH\_RSA\_DES\_168\_SHA, SSL\_RSA\_AES\_128\_SHA, SSL\_RSA\_AES\_256\_SHA, SSL\_RSA\_DES\_168\_SHA, SSL\_RSA\_RC4\_128\_MD5, SSL\_RSA\_RC4\_128\_SHA, ssl\_test(), and USAGE.

### 13.155.2.3 void my\_debug (void \* ctx, int level, char \* str)

Definition at line 113 of file ssl\_test.c.

### 13.155.2.4 static int ssl\_test (struct options \* opt) [static]

Definition at line 122 of file ssl\_test.c.

References options::buffer\_size, client\_fd, options::command, COMMAND\_READ, COMMAND\_WRITE, options::conn\_timeout, DFL\_FORCE\_CIPHER, dhm\_G, dhm\_P, options::force\_cipher, get\_timer(), havege\_init(), havege\_rand(), options::iomode, IOMODE\_NONBLOCK, lcppm5(), options::max\_bytes, my\_debug(), net\_accept(), net\_bind(), net\_close(), net\_connect(), net\_recv(), net\_send(), net\_set\_nonblock(), \_x509\_cert::next, options::opmode, OPMODE\_CLIENT, OPMODE\_SERVER, rsa\_free(), server\_fd, options::server\_name, options::server\_port, options::session\_lifetime, options::session\_reuse, ssl\_close\_notify(), ssl\_default\_ciphers, ssl\_free(), ssl\_init(), SSL\_IS\_CLIENT, SSL\_IS\_SERVER, ssl\_read(), ssl\_set\_authmode(), ssl\_set\_bio(), ssl\_set\_ca\_chain(), ssl\_set\_ciphers(), ssl\_set\_dbg(), ssl\_set\_dh\_param(), ssl\_set\_endpoint(), ssl\_set\_own\_cert(), ssl\_set\_rng(), ssl\_set\_session(), SSL\_VERIFY\_NONE, ssl\_write(), test\_ca crt, test\_srv crt, test\_srv\_key, x509\_free(), x509parse\_crt(), x509parse\_key(), XYSSL\_ERR\_NET\_CONN\_RESET, XYSSL\_ERR\_NET\_TRY\_AGAIN, and XYSSL\_ERR\_SSL\_PEER\_CLOSE\_NOTIFY.

Referenced by main().

### 13.155.3 Variable Documentation

#### 13.155.3.1 `char* dhm_G = "4"`

Definition at line 63 of file `ssl_test.c`.

Referenced by `ssl_test()`.

#### 13.155.3.2 `char* dhm_P`

**Initial value:**

```
"E4004C1F94182000103D883A448B3F802CE4B44A83301270002C20D0321CFD00"  
"11CCEF784C26A400F43DFB901BCA7538F2C6B176001CF5A0FD16D2C48B1D0C1C"  
"F6AC8E1DA6BCC3B4E1F96B0564965300FFA1D0B601EB2800F489AA512C4B248C"  
"01F76949A60BB7F00A40B1EAB64BDD48E8A700D60B7F1200FA8E77B0A979DABF"
```

Definition at line 64 of file `ssl_test.c`.

Referenced by `ssl_test()`.

#### 13.155.3.3 `int server_fd = -1`

Definition at line 70 of file `ssl_test.c`.

## 13.156 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/util/mc\_genkey.c File Reference

```
#include "../security/interface.h"
```

### Functions

- void [print\\_usage](#) ()
- int [main](#) (int argc, char \*argv[])

#### 13.156.1 Function Documentation

##### 13.156.1.1 int main (int *argc*, char \* *argv*[])

Definition at line 19 of file mc\_genkey.c.

References [generate\\_RSA\\_keys\\_ciphertext\(\)](#), [generate\\_RSA\\_keys\\_plaintext\(\)](#), and [print\\_usage\(\)](#).

##### 13.156.1.2 void print\_usage ()

Definition at line 11 of file mc\_genkey.c.

Referenced by [main\(\)](#).

## 13.157 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/win32/EmbeddedCh.Net/EmbeddedCh.Net/Ch.cs File Reference

### Data Structures

- struct [EmbeddedCh::ChInfo\\_t](#)

### Namespaces

- namespace [EmbeddedCh](#)

### Enumerations

- enum [EmbeddedCh::ChType\\_t](#) {  
[EmbeddedCh::CH\\_UNDEFINETYPE](#), [EmbeddedCh::CH\\_CHARTYPE](#) = 10, [EmbeddedCh::CH\\_UCHARTYPE](#), [EmbeddedCh::CH\\_SHORTTYPE](#),  
[EmbeddedCh::CH\\_USHORTTYPE](#), [EmbeddedCh::CH\\_INTTYPE](#), [EmbeddedCh::CH\\_UINTTYPE](#), [EmbeddedCh::CH\\_LLINTTYPE](#),  
[EmbeddedCh::CH\\_ULLINTTYPE](#), [EmbeddedCh::CH\\_FLOATTYPE](#), [EmbeddedCh::CH\\_DOUBLETYPE](#), [EmbeddedCh::CH\\_LDOUBLETYPE](#),  
[EmbeddedCh::CH\\_COMPLEXTYPE](#), [EmbeddedCh::CH\\_LCOMPLEXTYPE](#),  
[EmbeddedCh::CH\\_STRINGTYPE](#), [EmbeddedCh::CH\\_FILETYPE](#),  
[EmbeddedCh::CH\\_VOIDTYPE](#), [EmbeddedCh::CH\\_PROCTYPE](#), [EmbeddedCh::CH\\_STRUCTTYPE](#), [EmbeddedCh::CH\\_CLASSTYPE](#),  
[EmbeddedCh::CH\\_UNIONTYPE](#), [EmbeddedCh::CH\\_ENUMTYPE](#), [EmbeddedCh::CH\\_CARRAYTYPE](#) = 80, [EmbeddedCh::CH\\_CARRAYPTRTYPE](#),  
[EmbeddedCh::CH\\_CARRAYVLATTYPE](#), [EmbeddedCh::CH\\_CHARRAYTYPE](#),  
[EmbeddedCh::CH\\_CHARRAYPTRTYPE](#), [EmbeddedCh::CH\\_CHARRAYVLATTYPE](#),  
[EmbeddedCh::CH\\_NULLTYPE](#) = 100, [EmbeddedCh::CH\\_VOIDPTRTYPE](#), [EmbeddedCh::CH\\_CHARPTRTYPE](#), [EmbeddedCh::CH\\_UCHARPTRTYPE](#),  
[EmbeddedCh::CH\\_SHORTPTRTYPE](#), [EmbeddedCh::CH\\_USHORTPTRTYPE](#),  
[EmbeddedCh::CH\\_INTPTRTYPE](#), [EmbeddedCh::CH\\_UINTPTRTYPE](#),  
[EmbeddedCh::CH\\_LLINTPTRTYPE](#), [EmbeddedCh::CH\\_ULLINTPTRTYPE](#), [EmbeddedCh::CH\\_FLOATPTRTYPE](#), [EmbeddedCh::CH\\_DOUBLEPTRTYPE](#),  
[EmbeddedCh::CH\\_LDOUBLEPTRTYPE](#), [EmbeddedCh::CH\\_COMPLEXPTRTYPE](#),  
[EmbeddedCh::CH\\_LCOMPLEXPTRTYPE](#), [EmbeddedCh::CH\\_STRINGPTRTYPE](#),  
[EmbeddedCh::CH\\_PROCPTRTYPE](#), [EmbeddedCh::CH\\_FILEPTRTYPE](#), [EmbeddedCh::CH\\_STRUCTPTRTYPE](#), [EmbeddedCh::CH\\_CLASSPTRTYPE](#),  
[EmbeddedCh::CH\\_UNIONPTRTYPE](#), [EmbeddedCh::CH\\_ENUMPTRTYPE](#), [EmbeddedCh::CH\\_VOIDPTR2TYPE](#) = 200, [EmbeddedCh::CH\\_CHARPTR2TYPE](#),  
[EmbeddedCh::CH\\_UCHARPTR2TYPE](#), [EmbeddedCh::CH\\_SHORTPTR2TYPE](#),  
[EmbeddedCh::CH\\_USHORTPTR2TYPE](#), [EmbeddedCh::CH\\_INTPTR2TYPE](#),  
[EmbeddedCh::CH\\_UINTPTR2TYPE](#), [EmbeddedCh::CH\\_LLINTPTR2TYPE](#), [EmbeddedCh::CH\\_ULLINTPTR2TYPE](#), [EmbeddedCh::CH\\_FLOATPTR2TYPE](#),

---

[EmbeddedCh::CH\\_DOUBLEPTR2TYPE](#), [EmbeddedCh::CH\\_LDOUBLEPTR2TYPE](#),  
[EmbeddedCh::CH\\_COMPLEXPTR2TYPE](#), [EmbeddedCh::CH\\_LCOMPLEXPTR2TYPE](#),  
[EmbeddedCh::CH\\_STRINGPTR2TYPE](#), [EmbeddedCh::CH\\_FILEPTR2TYPE](#),  
[EmbeddedCh::CH\\_STRUCTPTR2TYPE](#), [EmbeddedCh::CH\\_CLASSPTR2TYPE](#),  
[EmbeddedCh::CH\\_UNIONPTR2TYPE](#), [EmbeddedCh::CH\\_ENUMPTR2TYPE](#) }  
• `enum EmbeddedCh::ChRetVal { EmbeddedCh::CH_OK = 0, EmbeddedCh::CH_ERROR = -1,  
EmbeddedCh::CH_ABORT = 1 }`

## 13.158 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/win32/EmbeddedCh.Net/EmbeddedCh.Net/ChInterp.cs

### File Reference

#### Data Structures

- class [EmbeddedCh::ChInterp](#)

#### Namespaces

- namespace [EmbeddedCh](#)



13.159 /home/dko/Projects/mobilec/tags/MobileC-  
v1.10.10/src/win32/EmbeddedCh.Net/EmbeddedCh.Net/ChUserDefinedTag.cs File

Reference

949

~~13.159 /home/dko/Projects/mobilec/tags/MobileC-~~

~~v1.10.10/src/win32/EmbeddedCh.Net/EmbeddedCh.Net/ChUserDefinedTa~~

## File Reference

### Data Structures

- class [EmbeddedCh::ChUserDefinedTag](#)

### Namespaces

- namespace [EmbeddedCh](#)

## 13.160 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/win32/EmbeddedCh.Net/EmbeddedCh.Net/ChVaList.cs

### File Reference

#### Data Structures

- class [EmbeddedCh::ChVaList](#)

#### Namespaces

- namespace [EmbeddedCh](#)

## **13.161 /home/dko/Projects/mobilec/tags/MobileC- v1.10.10/src/win32/EmbeddedCh.Net/EmbeddedCh.Net/EmbedCh.cs File Reference**

### **Data Structures**

- struct [EmbeddedCh::ChOptions\\_t](#)
- struct [EmbeddedCh::ChBlock\\_t](#)
- struct [EmbeddedCh::ChUserDefinedInfo\\_t](#)
- struct [EmbeddedCh::ChMemInfo\\_t](#)

### **Namespaces**

- namespace [EmbeddedCh](#)

### **Enumerations**

- enum [EmbeddedCh::ChFuncType\\_t](#) {  
    [EmbeddedCh::CH\\_NOTFUNCTYPE](#),   [EmbeddedCh::CH\\_FUNCTYPE](#),   [EmbeddedCh::CH\\_-  
FUNCPROTOTYPE](#), [EmbeddedCh::CH\\_FUNCPTRTYPE](#),  
    [EmbeddedCh::CH\\_FUNCMEMBERTYPE](#),                   [EmbeddedCh::CH\\_FUNCCONSTYPE](#),  
    [EmbeddedCh::CH\\_FUNCDESTTYPE](#) }  
• enum [EmbeddedCh::ChVarType\\_t](#) { [EmbeddedCh::CH\\_NOTVARTYPE](#), [EmbeddedCh::CH\\_-  
GLOBALVARTYPE](#), [EmbeddedCh::CH\\_LOCALVARTYPE](#) }  
• enum [EmbeddedCh::ChShellType](#) { [EmbeddedCh::CH\\_REGULARCH](#) = 0, [EmbeddedCh::CH\\_-  
SAFECH](#) = 1 }  
    *Ch shell type.*  
• enum [EmbeddedCh::ChFileDescriptor](#) {   [EmbeddedCh::STDIN\\_FILENO](#)     =   0,  
    [EmbeddedCh::STDOUT\\_FILENO](#) = 1, [EmbeddedCh::STDERR\\_FILENO](#) = 2 }  
• enum [EmbeddedCh::ChCallbackMask](#) {  
    [EmbeddedCh::CH\\_MASKNONE](#)   = 0X0000, [EmbeddedCh::CH\\_MASKCALL](#)   = 0X0001,  
    [EmbeddedCh::CH\\_MASKRET](#) = 0X0002, [EmbeddedCh::CH\\_MASKBLOCK](#) = 0X0004,  
    [EmbeddedCh::CH\\_MASKEND](#)   = 0X0008, [EmbeddedCh::CH\\_MASKLINE](#)   = 0X0010,  
    [EmbeddedCh::CH\\_MASKCOUNT](#) = 0X0020, [EmbeddedCh::CH\\_MASKABORT](#) = 0X0040  
    }  
• enum [EmbeddedCh::ChFileDescriptor](#) {   [EmbeddedCh::STDIN\\_FILENO](#)     =   0,  
    [EmbeddedCh::STDOUT\\_FILENO](#) = 1, [EmbeddedCh::STDERR\\_FILENO](#) = 2 }  
• enum [EmbeddedCh::ChCallbackMask](#) {  
    [EmbeddedCh::CH\\_MASKNONE](#)   = 0X0000, [EmbeddedCh::CH\\_MASKCALL](#)   = 0X0001,  
    [EmbeddedCh::CH\\_MASKRET](#) = 0X0002, [EmbeddedCh::CH\\_MASKBLOCK](#) = 0X0004,  
    [EmbeddedCh::CH\\_MASKEND](#)   = 0X0008, [EmbeddedCh::CH\\_MASKLINE](#)   = 0X0010,  
    [EmbeddedCh::CH\\_MASKCOUNT](#) = 0X0020, [EmbeddedCh::CH\\_MASKABORT](#) = 0X0040  
    }

## 13.162 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/win32/EmbeddedCh.Net/EmbeddedCh.Net/Properties/AssemblyInfo.cs

### File Reference

#### Variables

- using System. [Reflection](#)

#### 13.162.1 Variable Documentation

##### 13.162.1.1 using System. Reflection

Definition at line 1 of file AssemblyInfo.cs.

## File Reference

### Variables

- using System. [Reflection](#)

### 13.163.1 Variable Documentation

#### 13.163.1.1 using System. Reflection

Definition at line 1 of file AssemblyInfo.cs.

## **13.164    /home/dko/Projects/mobilec/tags/MobileC- v1.10.10/src/win32/LibMC.net/LibMC/Properties/AssemblyInfo.cs File Reference**

### **Variables**

- using System. [Reflection](#)

### **13.164.1    Variable Documentation**

#### **13.164.1.1    using System. Reflection**

Definition at line 1 of file AssemblyInfo.cs.

13.165 /home/dko/Projects/mobilec/tags/MobileC-  
v1.10.10/src/win32/EmbeddedCh.Net/Program1/Program.cs File

Reference

955

## **13.165 /home/dko/Projects/mobilec/tags/MobileC- v1.10.10/src/win32/EmbeddedCh.Net/Program1/Program.cs File Reference**

### **Data Structures**

- class [Program1::Program](#)

### **Namespaces**

- namespace [Program1](#)

## 13.166 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/win32/LibMC.net/LibMC/MCAclMessage.cs

### File Reference

#### Data Structures

- class [LibMC::MCAclMessage](#)  
*Encapsulates ACL messages in the Mobile-C library.*

#### Namespaces

- namespace [LibMC](#)  
*Namespace for the .NET wrapper for Mobile-C.*

#### 13.166.1 Detailed Description

Defines the MCAclMessage object and its member functions.

Definition in file [MCAclMessage.cs](#).



13.167

/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/win32/LibMC.net/LibMC/MCAgency.cs

File Reference

957

**13.167** /home/dko/Projects/mobilec/tags/MobileC-

v1.10.10/src/win32/LibMC.net/LibMC/MCAgency.cs

**File**

**Reference**

## Data Structures

- class [LibMC::MCAgency](#)  
*Wrapper class for [MCAgency\\_t](#) structure.*
- class [LibMC::InvalidAgencyException](#)  
*Exception class for use with null agency pointers.*

## Namespaces

- namespace [LibMC](#)  
*Namespace for the .NET wrapper for Mobile-C.*

### 13.167.1 Detailed Description

Defines the MCAgency object and its member functions.

Definition in file [MCAgency.cs](#).

## 13.168 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/win32/LibMC.net/LibMC/MCAgent.cs File Reference

### Data Structures

- class [LibMC::MCAgent](#)  
*Wrapper class for MCAgent\_t structure.*
- class [LibMC::InvalidAgentException](#)  
*Exception class for use with null agent pointers.*

### Namespaces

- namespace [LibMC](#)  
*Namespace for the .NET wrapper for Mobile-C.*

#### 13.168.1 Detailed Description

Defines the MCAgent object and its member functions.

Definition in file [MCAgent.cs](#).

13.169

/home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/win32/LibMC.net/LibMC/MCExports.cs

File Reference

959

**13.169** /home/dko/Projects/mobilec/tags/MobileC-

**v1.10.10/src/win32/LibMC.net/LibMC/MCExports.cs    File**

## Reference

### Data Structures

- class [LibMC::MCAgency](#)  
*Wrapper class for [MCAgency\\_t](#) structure.*
- struct [LibMC::MCAgency::MCAgency\\_t](#)
- struct [LibMC::MCAgency::MCAgencyOptions\\_t](#)
- struct [LibMC::MCAgency::ChOptions\\_t](#)  
*ChOptions structures.*

### Namespaces

- namespace [LibMC](#)  
*Namespace for the .NET wrapper for Mobile-C.*

#### 13.169.1 Detailed Description

Imports functions, structs, and enums from the Mobile-C library.

Definition in file [MCExports.cs](#).

## 13.170 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/win32/LibMC.net/LibMC/Properties/Settings.Designer.cs

### File Reference

#### Data Structures

- class [LibMC::Properties::Settings](#)

#### Namespaces

- namespace [LibMC::Properties](#)  
*Namespace for the .NET wrapper properties class.*

## 13.171 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/win32/LibMC.net/LibMC/Settings.cs File Reference

### Data Structures

- class [LibMC::Properties::Settings](#)

### Namespaces

- namespace [LibMC::Properties](#)  
*Namespace for the .NET wrapper properties class.*

## 13.172 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/winconfig.h File Reference

### Defines

- #define `PACKAGE_STRING` "MobileC V1.10.0"
- #define `PACKAGE_VERSION` "1.10.0"
- #define `strtok_r(buf, delim, save_ptr)` `strtok( buf, delim )`

### 13.172.1 Define Documentation

#### 13.172.1.1 #define `PACKAGE_STRING` "MobileC V1.10.0"

Definition at line 4 of file winconfig.h.

Referenced by `mtp_http_InitializeFromConnection()`.

#### 13.172.1.2 #define `PACKAGE_VERSION` "1.10.0"

Definition at line 5 of file winconfig.h.

Referenced by `mtp_http_ComposeMessage()`, `mtp_http_CreateMessage()`, and `udplisten_Thread()`.

#### 13.172.1.3 #define `strtok_r(buf, delim, save_ptr)` `strtok( buf, delim )`

Definition at line 6 of file winconfig.h.

Referenced by `agent_xml_parse__fill_row_data()`, `message_InitializeFromAgent()`, `message_send_Thread()`, and `message_xml_parse__message()`.

## 13.173 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/xml\_compose.c File Reference

```
#include "config.h"
#include <mxml.h>
#include "include/agent.h"
#include "include/xml_compose.h"
#include "include/xml_helper.h"
```

### Functions

- [mxml\\_node\\_t \\* agent\\_xml\\_compose \(agent\\_p agent\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_gaf\\_message \(agent\\_p agent\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_message \(agent\\_p agent\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_mobile\\_agent \(agent\\_p agent\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_agent\\_data \(agent\\_p agent\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_name \(agent\\_p agent\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_owner \(agent\\_p agent\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_home \(agent\\_p agent\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_wg\\_code \(agent\\_p agent\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_tasks \(agent\\_p agent\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_task \(agent\\_p agent, int index\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_data \(agent\\_p agent, int index, interpreter\\_variable\\_data\\_t \\*interp\\_variable\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_agent\\_code \(agent\\_p agent, int index\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_row \(interpreter\\_variable\\_data\\_t \\*interp\\_variable, int index\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_create\\_row\\_nodes \(void \\*data, int index, int \\*extent, ChType\\_t type, int dim, int extent\\_index\)](#)

### 13.173.1 Function Documentation

#### 13.173.1.1 [mxml\\_node\\_t\\* agent\\_xml\\_compose \(agent\\_p agent\)](#)

Definition at line 46 of file `xml_compose.c`.

References `agent_xml_compose__gaf_message()`, `MXML_ADD_AFTER`, `MXML_ADD_TO_PARENT`, `MXML_NO_CALLBACK`, `mxmlAdd()`, `mxmlLoadString()`, and `node`.

Referenced by `message_InitializeFromAgent()`.

#### 13.173.1.2 [mxml\\_node\\_t\\* agent\\_xml\\_compose\\_\\_agent\\_code \(agent\\_p agent, int index\)](#)

Definition at line 521 of file `xml_compose.c`.

References `agent_datastate_s::agent_code_ids`, `agent_datastate_s::agent_codes`, `agent_s::datastate`, `MXML_NO_PARENT`, `mxmlElementSetAttr()`, `mxmlNewElement()`, `node`, and `xml_new_cdata()`.

Referenced by `agent_xml_compose__tasks()`.

**13.173.1.3 mxmml\_node\_t\* agent\_xml\_compose\_\_agent\_data (agent\_p agent)**

Definition at line 150 of file xml\_compose.c.

References agent\_xml\_compose\_\_home(), agent\_xml\_compose\_\_name(), agent\_xml\_compose\_\_owner(), agent\_xml\_compose\_\_tasks(), agent\_xml\_compose\_\_wg\_code(), MXML\_ADD\_AFTER, mxmmlAdd(), mxmmlNewElement(), and node.

Referenced by agent\_xml\_compose\_\_mobile\_agent().

**13.173.1.4 mxmml\_node\_t\* agent\_xml\_compose\_\_create\_row\_nodes (void \* data, int index, int \* extent, ChType\_t type, int dim, int extent\_index)**

Definition at line 572 of file xml\_compose.c.

References agent\_xml\_compose\_\_create\_row\_nodes(), buf, CH\_DATATYPE\_SIZE, CH\_DATATYPE\_VALUE\_STRING, MXML\_ADD\_AFTER, MXML\_ADD\_TO\_PARENT, MXML\_NO\_PARENT, mxmmlAdd(), mxmmlElementSetAttr(), mxmmlNewElement(), mxmmlNewText(), node, and size.

Referenced by agent\_xml\_compose\_\_create\_row\_nodes(), and agent\_xml\_compose\_\_row().

**13.173.1.5 mxmml\_node\_t\* agent\_xml\_compose\_\_data (agent\_p agent, int index, interpreter\_variable\_data\_t \* interp\_variable)**

Definition at line 447 of file xml\_compose.c.

References agent\_xml\_compose\_\_row(), interpreter\_variable\_data\_s::array\_dim, buf, CH\_DATATYPE\_STRING, CH\_DATATYPE\_VALUE\_STRING, interpreter\_variable\_data\_s::data, interpreter\_variable\_data\_s::data\_type, MXML\_ADD\_AFTER, mxmmlAdd(), mxmmlElementSetAttr(), mxmmlNewElement(), interpreter\_variable\_data\_s::name, and node.

Referenced by agent\_xml\_compose\_\_task().

**13.173.1.6 mxmml\_node\_t\* agent\_xml\_compose\_\_gaf\_message (agent\_p agent)**

Definition at line 66 of file xml\_compose.c.

References agent\_xml\_compose\_\_message(), MXML\_ADD\_AFTER, mxmmlAdd(), mxmmlNewElement(), and node.

Referenced by agent\_xml\_compose().

**13.173.1.7 mxmml\_node\_t\* agent\_xml\_compose\_\_home (agent\_p agent)**

Definition at line 253 of file xml\_compose.c.

References agent\_s::home, mxmmlNewElement(), mxmmlNewText(), and node.

Referenced by agent\_xml\_compose\_\_agent\_data().

**13.173.1.8 mxmml\_node\_t\* agent\_xml\_compose\_\_message (agent\_p agent)**

Definition at line 85 of file xml\_compose.c.



References `agent_s::agent_type`, `agent_xml_compose__mobile_agent()`, `MC_LOCAL_AGENT`, `MC_REMOTE_AGENT`, `MC_RETURN_AGENT`, `MXML_ADD_AFTER`, `mxmlAdd()`, `mxmlElementSetAttr()`, `mxmlNewElement()`, and `node`.

Referenced by `agent_xml_compose__gaf_message()`.

#### 13.173.1.9 `mxml_node_t* agent_xml_compose__mobile_agent (agent_p agent)`

Definition at line 129 of file `xml_compose.c`.

References `agent_xml_compose__agent_data()`, `MXML_ADD_AFTER`, `mxmlAdd()`, `mxmlNewElement()`, and `node`.

Referenced by `agent_xml_compose__message()`.

#### 13.173.1.10 `mxml_node_t* agent_xml_compose__name (agent_p agent)`

Definition at line 221 of file `xml_compose.c`.

References `mxmlNewElement()`, `mxmlNewText()`, `agent_s::name`, and `node`.

Referenced by `agent_xml_compose__agent_data()`.

#### 13.173.1.11 `mxml_node_t* agent_xml_compose__owner (agent_p agent)`

Definition at line 237 of file `xml_compose.c`.

References `mxmlNewElement()`, `mxmlNewText()`, `node`, and `agent_s::owner`.

Referenced by `agent_xml_compose__agent_data()`.

#### 13.173.1.12 `mxml_node_t* agent_xml_compose__row (interpreter_variable_data_t * interp_variable, int index)`

Definition at line 549 of file `xml_compose.c`.

References `agent_xml_compose__create_row_nodes()`, `interpreter_variable_data_s::array_dim`, `interpreter_variable_data_s::array_extent`, `interpreter_variable_data_s::data`, `interpreter_variable_data_s::data_type`, and `node`.

Referenced by `agent_xml_compose__data()`.

#### 13.173.1.13 `mxml_node_t* agent_xml_compose__task (agent_p agent, int index)`

Definition at line 342 of file `xml_compose.c`.

References `agent_task_s::agent_return_data`, `agent_task_s::agent_variable_list`, `agent_xml_compose__data()`, `buf`, `agent_task_s::code_id`, `agent_s::datastate`, `MXML_ADD_AFTER`, `mxmlAdd()`, `mxmlElementSetAttr()`, `mxmlNewElement()`, `node`, `agent_task_s::persistent`, `agent_datastate_s::persistent`, `agent_task_s::server_name`, `agent_datastate_s::tasks`, and `agent_task_s::var_name`.

Referenced by `agent_xml_compose__tasks()`.

**13.173.1.14   mxml\_node\_t\* agent\_xml\_compose\_\_tasks (agent\_p *agent*)**

Definition at line 285 of file xml\_compose.c.

References agent\_xml\_compose\_\_agent\_code(), agent\_xml\_compose\_\_task(), buf, agent\_s::datastate, MXML\_ADD\_AFTER, mxmlAdd(), mxmlElementSetAttr(), mxmlNewElement(), node, agent\_datastate\_s::number\_of\_tasks, and agent\_datastate\_s::task\_progress.

Referenced by agent\_xml\_compose\_\_agent\_data().

**13.173.1.15   mxml\_node\_t\* agent\_xml\_compose\_\_wg\_code (agent\_p *agent*)**

Definition at line 269 of file xml\_compose.c.

References mxmlNewElement(), mxmlNewText(), node, and agent\_s::wg\_code.

Referenced by agent\_xml\_compose\_\_agent\_data().

## 13.174 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/xml\_helper.c File Reference

```
#include "config.h"
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <mxml.h>
#include "include/xml_helper.h"
#include "include/macros.h"
```

### Functions

- [mxml\\_node\\_t \\* xml\\_find\\_sibling](#) (const [mxml\\_node\\_t](#) \**node*, const char \**sibling\_name*)
- char \* [xml\\_get\\_cdata](#) (const [mxml\\_node\\_t](#) \**node*)
- [mxml\\_node\\_t \\* xml\\_get\\_child](#) (const [mxml\\_node\\_t](#) \**node*, const char \**child\_name*, [int](#) *descend*)
- [mxml\\_node\\_t \\* xml\\_get\\_deep\\_child](#) (const [mxml\\_node\\_t](#) \**parent*, const char \*\**child\_path*)
- [mxml\\_node\\_t \\* xml\\_get\\_next\\_element](#) (const [mxml\\_node\\_t](#) \**node*)
- char \* [xml\\_get\\_text](#) (const [mxml\\_node\\_t](#) \**node*)
- const char \* [xml\\_get\\_element\\_name](#) (const [mxml\\_node\\_t](#) \**node*)
- [mxml\\_node\\_t \\* xml\\_new\\_cdata](#) ([mxml\\_node\\_t](#) \**parent*, const char \**text*)
- const char \* [whitespace\\_cb](#) ([mxml\\_node\\_t](#) \**node*, [int](#) *where*)

### 13.174.1 Function Documentation

#### 13.174.1.1 const char\* whitespace\_cb (mxml\_node\_t \* *node*, int *where*)

Definition at line 254 of file `xml_helper.c`.

References `MXML_WS_AFTER_CLOSE`, and `MXML_WS_BEFORE_OPEN`.

Referenced by `main()`.

#### 13.174.1.2 mxml\_node\_t\* xml\_find\_sibling (const mxml\_node\_t \* *node*, const char \* *sibling\_name*)

Definition at line 54 of file `xml_helper.c`.

References `MXML_NO_DESCEND`, `mxmlFindElement()`, `mxml_node_s::parent`, and `xml_get_element_name()`.

#### 13.174.1.3 char\* xml\_get\_cdata (const mxml\_node\_t \* *node*)

Definition at line 76 of file `xml_helper.c`.

References `buf`, `CHECK_NULL`, `MXML_ELEMENT`, `mxml_node_s::type`, and `xml_get_element_name()`.

Referenced by `xml_get_text()`.

**13.174.1.4 mxml\_node\_t\* xml\_get\_child (const mxml\_node\_t \* node, const char \* child\_name, int descend)**

Definition at line 109 of file xml\_helper.c.

References mxmlFindElement().

Referenced by agent\_return\_xml\_parse(), agent\_xml\_parse\_\_agent\_data(), agent\_xml\_parse\_\_data(), agent\_xml\_parse\_\_mobile\_agent(), message\_xml\_parse(), message\_xml\_parse\_\_message(), and xml\_get\_deep\_child().

**13.174.1.5 mxml\_node\_t\* xml\_get\_deep\_child (const mxml\_node\_t \* parent, const char \*\* child\_path)**

Definition at line 128 of file xml\_helper.c.

References MXML\_NO\_DESCEND, node, and xml\_get\_child().

**13.174.1.6 const char\* xml\_get\_element\_name (const mxml\_node\_t \* node)**

Definition at line 222 of file xml\_helper.c.

References mxml\_value\_u::element, MXML\_ELEMENT, mxml\_value\_s::name, mxml\_node\_s::type, and mxml\_node\_s::value.

Referenced by agent\_xml\_parse\_\_data(), agent\_xml\_parse\_\_mobile\_agent(), agent\_xml\_parse\_\_row(), message\_xml\_parse(), xml\_find\_sibling(), and xml\_get\_cdata().

**13.174.1.7 mxml\_node\_t\* xml\_get\_next\_element (const mxml\_node\_t \* node)**

Definition at line 142 of file xml\_helper.c.

References MXML\_ELEMENT, mxml\_node\_s::next, and mxml\_node\_s::type.

**13.174.1.8 char\* xml\_get\_text (const mxml\_node\_t \* node)**

Definition at line 160 of file xml\_helper.c.

References CHECK\_NULL, mxml\_node\_s::child, mxml\_value\_u::element, MXML\_ELEMENT, MXML\_TEXT, mxml\_value\_s::name, mxml\_node\_s::next, mxml\_text\_s::string, mxml\_value\_u::text, mxml\_node\_s::type, mxml\_node\_s::value, and xml\_get\_cdata().

Referenced by agent\_xml\_parse\_\_agent\_code(), agent\_xml\_parse\_\_home(), agent\_xml\_parse\_\_name(), agent\_xml\_parse\_\_owner(), agent\_xml\_parse\_\_sender(), and agent\_xml\_parse\_\_wg\_code().

**13.174.1.9 mxml\_node\_t\* xml\_new\_cdata (mxml\_node\_t \* parent, const char \* text)**

Definition at line 235 of file xml\_helper.c.

References CHECK\_NULL, mxmlNewElement(), and node.

Referenced by agent\_xml\_compose\_\_agent\_code().

## 13.175 /home/dko/Projects/mobilec/tags/MobileC-v1.10.10/src/xml\_parser.c File Reference

```
#include <mxml.h>
#include <string.h>
#include <stdlib.h>
#include "config.h"
#include "include/interpreter_variable_data.h"
#include "include/message.h"
#include "include/xml_parser.h"
#include "include/xml_helper.h"
```

### Defines

- `#define \_XOPEN\_SOURCE 600`

### Functions

- `error\_code\_t agent\_xml\_parse (agent\_p agent)`
- `error\_code\_t agent\_xml\_parse\_\_mobile\_agent (agent\_p agent, xml\_parser\_p xml_parser)`
- `error\_code\_t agent\_xml\_parse\_\_agent\_data (agent\_p agent, xml\_parser\_p xml_parser)`
- `error\_code\_t agent\_xml\_parse\_\_name (agent\_p agent, xml\_parser\_p xml_parser)`
- `error\_code\_t agent\_xml\_parse\_\_owner (agent\_p agent, xml\_parser\_p xml_parser)`
- `error\_code\_t agent\_xml\_parse\_\_home (agent\_p agent, xml\_parser\_p xml_parser)`
- `error\_code\_t agent\_xml\_parse\_\_sender (agent\_p agent, xml\_parser\_p xml_parser)`
- `error\_code\_t agent\_xml\_parse\_\_wg\_code (agent\_p agent, xml\_parser\_p xml_parser)`
- `error\_code\_t agent\_xml\_parse\_\_tasks (agent\_p agent, xml\_parser\_p xml_parser)`
- `error\_code\_t agent\_xml\_parse\_\_task (agent\_p agent, xml\_parser\_p xml_parser, int index)`
- `error\_code\_t agent\_xml\_parse\_\_data (agent\_p agent, xml\_parser\_p xml_parser, int index)`
- `error\_code\_t agent\_xml\_parse\_\_row (interpreter\_variable\_data\_t *interp_variable, xml\_parser\_p xml_parser, int index)`
- `void agent\_xml\_parse\_\_fill\_row\_data (void *data, ChType\_t type, int *extent, const mxml\_node\_t *node, int *index)`
- `error\_code\_t agent\_xml\_parse\_\_agent\_code (agent\_p agent, int index, xml\_parser\_p xml_parser)`
- `error\_code\_t agent\_return\_xml\_parse (agent\_p agent)`
- `error\_code\_t message\_xml\_parse (message\_p message)`
- `error\_code\_t message\_xml\_parse\_\_message (message\_p message, xml\_parser\_p xml_parser)`

### 13.175.1 Define Documentation

#### 13.175.1.1 `#define \_XOPEN\_SOURCE 600`

Definition at line 38 of file `xml_parser.c`.

## 13.175.2 Function Documentation

### 13.175.2.1 `error_code_t agent_return_xml_parse (agent_p agent)`

Definition at line 913 of file `xml_parser.c`.

References `agent_xml_parse__home()`, `agent_xml_parse__name()`, `agent_xml_parse__owner()`, `agent_xml_parse__tasks()`, `agent_s::datastate`, `MC_SUCCESS`, `xml_get_child()`, and `agent_datastate_s::xml_root`.

### 13.175.2.2 `error_code_t agent_xml_parse (agent_p agent)`

Definition at line 52 of file `xml_parser.c`.

References `agent_xml_parse__mobile_agent()`, `agent_s::datastate`, `MC_SUCCESS`, and `agent_datastate_s::xml_agent_root`.

Referenced by `agent_initialize()`.

### 13.175.2.3 `error_code_t agent_xml_parse__agent_code (agent_p agent, int index, xml_parser_p xml_parser)`

Definition at line 872 of file `xml_parser.c`.

References `agent_datastate_s::agent_code`, `agent_datastate_s::agent_code_ids`, `agent_datastate_s::agent_codes`, `agent_task_s::code_id`, `agent_s::datastate`, `MC_SUCCESS`, `mxmlElementGetAttr()`, `agent_datastate_s::number_of_tasks`, `agent_datastate_s::task_progress`, `agent_datastate_s::tasks`, and `xml_get_text()`.

Referenced by `agent_xml_parse__tasks()`.

### 13.175.2.4 `error_code_t agent_xml_parse__agent_data (agent_p agent, xml_parser_p xml_parser)`

Definition at line 93 of file `xml_parser.c`.

References `agent_xml_parse__home()`, `agent_xml_parse__name()`, `agent_xml_parse__owner()`, `agent_xml_parse__sender()`, `agent_xml_parse__tasks()`, `agent_xml_parse__wg_code()`, `MC_ERR_PARSE`, `MC_SUCCESS`, and `xml_get_child()`.

Referenced by `agent_xml_parse__mobile_agent()`.

### 13.175.2.5 `error_code_t agent_xml_parse__data (agent_p agent, xml_parser_p xml_parser, int index)`

Definition at line 538 of file `xml_parser.c`.

References `agent_task_s::agent_return_data`, `agent_task_s::agent_variable_list`, `agent_xml_parse__row()`, `interpreter_variable_data_s::array_dim`, `CH_DATATYPE_SIZE`, `CH_DATATYPE_STR_TO_VAL`, `CH_STRING_DATATYPE`, `interpreter_variable_data_s::data`, `interpreter_variable_data_s::data_type`, `agent_s::datastate`, `interpreter_variable_data_New()`, `MC_ERR_PARSE`, `MC_SUCCESS`, `mxmlElementGetAttr()`, `interpreter_variable_data_s::name`, `mxml_node_s::parent`, `agent_task_s::persistent`, `agent_datastate_s::tasks`, `xml_get_child()`, and `xml_get_element_name()`.

Referenced by `agent_xml_parse__task()`.

**13.175.2.6 void agent\_xml\_parse\_\_fill\_row\_data (void \* *data*, ChType\_t *type*, int \* *extent*, const mxml\_node\_t \* *node*, int \* *index*)**

Definition at line 751 of file xml\_parser.c.

References agent\_xml\_parse\_\_fill\_row\_data(), buf, CH\_DATATYPE\_SIZE, mxml\_node\_s::child, MXML\_DESCEND\_FIRST, MXML\_ELEMENT, MXML\_TEXT, mxmlFindElement(), mxml\_text\_s::string, strtok\_r, mxml\_value\_u::text, mxml\_node\_s::type, and mxml\_node\_s::value.

Referenced by agent\_xml\_parse\_\_fill\_row\_data(), and agent\_xml\_parse\_\_row().

**13.175.2.7 error\_code\_t agent\_xml\_parse\_\_home (agent\_p *agent*, xml\_parser\_p *xml\_parser*)**

Definition at line 219 of file xml\_parser.c.

References CHECK\_NULL, agent\_s::home, MC\_SUCCESS, and xml\_get\_text().

Referenced by agent\_return\_xml\_parse(), and agent\_xml\_parse\_\_agent\_data().

**13.175.2.8 error\_code\_t agent\_xml\_parse\_\_mobile\_agent (agent\_p *agent*, xml\_parser\_p *xml\_parser*)**

Definition at line 65 of file xml\_parser.c.

References agent\_xml\_parse\_\_agent\_data(), MC\_ERR\_PARSE, xml\_get\_child(), and xml\_get\_element\_name().

Referenced by agent\_xml\_parse().

**13.175.2.9 error\_code\_t agent\_xml\_parse\_\_name (agent\_p *agent*, xml\_parser\_p *xml\_parser*)**

Definition at line 166 of file xml\_parser.c.

References CHECK\_NULL, MC\_ERR\_PARSE, MC\_SUCCESS, agent\_s::name, and xml\_get\_text().

Referenced by agent\_return\_xml\_parse(), and agent\_xml\_parse\_\_agent\_data().

**13.175.2.10 error\_code\_t agent\_xml\_parse\_\_owner (agent\_p *agent*, xml\_parser\_p *xml\_parser*)**

Definition at line 192 of file xml\_parser.c.

References CHECK\_NULL, MC\_SUCCESS, agent\_s::owner, and xml\_get\_text().

Referenced by agent\_return\_xml\_parse(), and agent\_xml\_parse\_\_agent\_data().

**13.175.2.11 error\_code\_t agent\_xml\_parse\_\_row (interpreter\_variable\_data\_t \* *interp\_variable*, xml\_parser\_p *xml\_parser*, int *index*)**

Definition at line 684 of file xml\_parser.c.

References agent\_xml\_parse\_\_fill\_row\_data(), interpreter\_variable\_data\_s::array\_dim, interpreter\_variable\_data\_s::array\_extent, CH\_DATATYPE\_SIZE, interpreter\_variable\_data\_s::data, interpreter\_variable\_data\_s::data\_type, MC\_SUCCESS, and xml\_get\_element\_name().

Referenced by agent\_xml\_parse\_\_data().

**13.175.2.12 error\_code\_t agent\_xml\_parse\_\_sender (agent\_p agent, xml\_parser\_p xml\_parser)**

Definition at line 245 of file xml\_parser.c.

References CHECK\_NULL, MC\_SUCCESS, agent\_s::sender, and xml\_get\_text().

Referenced by agent\_xml\_parse\_\_agent\_data().

**13.175.2.13 error\_code\_t agent\_xml\_parse\_\_task (agent\_p agent, xml\_parser\_p xml\_parser, int index)**

Definition at line 442 of file xml\_parser.c.

References agent\_xml\_parse\_\_data(), CHECK\_NULL, agent\_task\_s::code\_id, agent\_s::datastate, MC\_ERR\_PARSE, MC\_SUCCESS, MXML\_DESCEND\_FIRST, MXML\_NO\_DESCEND, mxmlElementGetAttr(), mxmlFindElement(), agent\_task\_s::persistent, agent\_task\_s::server\_name, agent\_datastate\_s::tasks, and agent\_task\_s::var\_name.

Referenced by agent\_xml\_parse\_\_tasks().

**13.175.2.14 error\_code\_t agent\_xml\_parse\_\_tasks (agent\_p agent, xml\_parser\_p xml\_parser)**

Definition at line 304 of file xml\_parser.c.

References agent\_datastate\_s::agent\_code, agent\_datastate\_s::agent\_code\_ids, agent\_datastate\_s::agent\_codes, agent\_task\_New(), agent\_xml\_parse\_\_agent\_code(), agent\_xml\_parse\_\_task(), buf, agent\_s::datastate, MC\_ERR\_PARSE, MXML\_DESCEND, MXML\_DESCEND\_FIRST, MXML\_NO\_DESCEND, mxmlElementGetAttr(), mxmlFindElement(), agent\_datastate\_s::number\_of\_tasks, agent\_datastate\_s::task\_progress, and agent\_datastate\_s::tasks.

Referenced by agent\_return\_xml\_parse(), and agent\_xml\_parse\_\_agent\_data().

**13.175.2.15 error\_code\_t agent\_xml\_parse\_\_wg\_code (agent\_p agent, xml\_parser\_p xml\_parser)**

Definition at line 273 of file xml\_parser.c.

References MC\_SUCCESS, agent\_s::wg\_code, and xml\_get\_text().

Referenced by agent\_xml\_parse\_\_agent\_data().

**13.175.2.16 error\_code\_t message\_xml\_parse (message\_p message)**

Definition at line 948 of file xml\_parser.c.

References MC\_ERR\_PARSE, message\_xml\_parse\_\_message(), MXML\_DESCEND, MXML\_NO\_DESCEND, mxmlFindElement(), xml\_get\_child(), xml\_get\_element\_name(), and message\_s::xml\_root.

Referenced by acc\_connection\_Thread(), message\_InitializeFromConnection(), and message\_InitializeFromString().

**13.175.2.17 error\_code\_t message\_xml\_parse\_\_message (message\_p message, xml\_parser\_p xml\_parser)**

Definition at line 1003 of file xml\_parser.c.



References `message_s::addr`, `buf`, `CHECK_NULL`, `ENCRYPTED_DATA`, `ENCRYPTION_INITIALIZE`, `FIPA_ACL`, `message_s::from_address`, `MC_ERR_PARSE`, `MC_SUCCESS`, `message_s::message_type`, `MOBILE_AGENT`, `mxmlElementGetAttr()`, `port`, `REQUEST_ENCRYPTION_INITIALIZE`, `RETURN_MSG`, `strtok_r`, `xml_get_child()`, and `message_s::xml_payload`.

Referenced by `message_xml_parse()`.



## **Chapter 14**

# **Example Documentation**

### **14.1 cs2ch/cs2chfunc.c**

C library example that allows agents to call CLI functions.

## 14.2 LibMCCConsole/Program.cs

Basic Mobile-C console demo program

## 14.3 LibMCCppEx/LibMCCppEx.cpp

Demonstrates using LibMC.NET from a VC++ program.

## 14.4 LibMCFipaTest/Program.cs

Mobile-C FIPA ACL message demo program.

## 14.5 LibMCGui/Form1.cs

Basic Mobile-C Windows Forms demo program

## 14.6 LibMCInterop/Program.cs

Demonstrates interfacing with a .dl file and calling agent functions.



## 14.7 LibMCMiscTest/Program.cs

Demonstrates miscellaneous Mobile-C functions.

## 14.8 LibMCVbEx/Form1.vb

Demonstrates using LibMC.NET from a VB program.