

# LibMC.NET Reference Manual

## 0.1

Generated by Doxygen 1.5.4

Tue Jun 17 12:10:29 2008



# Contents

<b>1</b>	<b>LibMC.NET</b>	<b>1</b>
1.1	Introduction . . . . .	1
<b>2</b>	<b>LibMC.NET Namespace Index</b>	<b>3</b>
2.1	LibMC.NET Package List . . . . .	3
<b>3</b>	<b>LibMC.NET Class Index</b>	<b>5</b>
3.1	LibMC.NET Class List . . . . .	5
<b>4</b>	<b>LibMC.NET File Index</b>	<b>7</b>
4.1	LibMC.NET File List . . . . .	7
<b>5</b>	<b>LibMC.NET Page Index</b>	<b>9</b>
5.1	LibMC.NET Related Pages . . . . .	9
<b>6</b>	<b>LibMC.NET Namespace Documentation</b>	<b>11</b>
6.1	Package LibMC . . . . .	11
6.2	Package LibMC.Properties . . . . .	12
<b>7</b>	<b>LibMC.NET Class Documentation</b>	<b>13</b>
7.1	LibMC.MCAclMessage Class Reference . . . . .	13
7.2	LibMC.MCAgency Class Reference . . . . .	19
7.3	LibMC.MCAgent Class Reference . . . . .	37
<b>8</b>	<b>LibMC.NET File Documentation</b>	<b>45</b>
8.1	/home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/MCAclMessage.cs File Reference . . . . .	45
8.2	/home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/MCAgency.cs File Reference . . . . .	46
8.3	/home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/MCAgent.cs File Reference . . . . .	47
8.4	/home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/MCEExports.cs File Reference . . . . .	48

<b>9</b>	<b>LibMC.NET Example Documentation</b>	<b>49</b>
9.1	LibMCConsole/Program.cs . . . . .	49
9.2	LibMCCppEx/LibMCCppEx.cpp . . . . .	50
9.3	LibMCFipaTest/Program.cs . . . . .	51
9.4	LibMCGui/Form1.cs . . . . .	52
9.5	LibMCMiscTest/Program.cs . . . . .	53
9.6	LibMCVbEx/Form1.vb . . . . .	54
<b>10</b>	<b>LibMC.NET Page Documentation</b>	<b>55</b>
10.1	MAclMessage . . . . .	55
10.2	Installing LibMC.NET . . . . .	56
10.3	Getting Started . . . . .	58
10.4	Using LibMC.NET . . . . .	59
10.5	Common Operations . . . . .	60
10.6	MAgency . . . . .	61
10.7	MAgent . . . . .	63
10.8	Todo List . . . . .	64
10.9	Bug List . . . . .	65

# Chapter 1

# LibMC.NET

**Author:**

Douglas P. Stark  
UC Davis and Sandia National Labs  
dpstark@sandia.gov

## 1.1 Introduction

Welcome to the LibMC.NET documentation. LibMC.NET is .NET class library that wraps Mobile-C in a layer to make it accessible to .NET applications. While the wrapper is continually under development, most of the features of Mobile-C are currently available to the programmer.



## Chapter 2

# LibMC.NET Namespace Index

### 2.1 LibMC.NET Package List

Here are the packages with brief descriptions (if available):

<a href="#">LibMC</a> (Namespace for the .NET wrapper for Mobile-C ) . . . . .	11
<a href="#">LibMC.Properties</a> (Namespace for the .NET wrapper properties class ) . . . . .	12





## Chapter 3

# LibMC.NET Class Index

### 3.1 LibMC.NET Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">LibMC.MCAclMessage</a> (Encapsulates ACL messages in the Mobile-C library ) . . . . .	<a href="#">13</a>
<a href="#">LibMC.MCAgency</a> (Wrapper class for MCAgency_t structure ) . . . . .	<a href="#">19</a>
<a href="#">LibMC.MCAgent</a> (Wrapper class for MCAgent_t structure ) . . . . .	<a href="#">37</a>



# Chapter 4

## LibMC.NET File Index

### 4.1 LibMC.NET File List

Here is a list of all documented files with brief descriptions:

<a href="#">/home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/MCAclMessage.cs</a>	45
<a href="#">/home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/MCAgency.cs</a>	46
<a href="#">/home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/MCAgent.cs</a>	47
<a href="#">/home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/MCExports.cs</a>	48
<a href="#">/home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/Settings.cs</a>	??



# Chapter 5

## LibMC.NET Page Index

### 5.1 LibMC.NET Related Pages

Here is a list of all related documentation pages:

Installing LibMC.NET . . . . .	56
Getting Started . . . . .	58
Using LibMC.NET . . . . .	59
Common Operations . . . . .	60
Todo List . . . . .	64
Bug List . . . . .	65



## Chapter 6

# LibMC.NET Namespace Documentation

### 6.1 Package LibMC

Namespace for the .NET wrapper for Mobile-C.

#### Classes

- class [MCAclMessage](#)  
*Encapsulates ACL messages in the Mobile-C library.*
- class [MCAgency](#)  
*Wrapper class for MCAgency\_t structure.*
- class [MCAgent](#)  
*Wrapper class for MCAgent\_t structure.*

#### Packages

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

#### 6.1.1 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.

## 6.2 Package LibMC.Properties

Namespace for the .NET wrapper properties class.

### 6.2.1 Detailed Description

Namespace for the .NET wrapper properties class.

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



## Chapter 7

# LibMC.NET Class Documentation

### 7.1 LibMC.MCAclMessage Class Reference

Encapsulates ACL messages in the Mobile-C library.

#### Public Types

- enum [MC\\_FipaPerformative\\_e](#)  
*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.*

### 7.1.1 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.

### 7.1.2 Member Enumeration Documentation

#### 7.1.2.1 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.

### 7.1.3 Constructor & Destructor Documentation

#### 7.1.3.1 LibMC.MCAclMessage.MCAclMessage ()

Default constructor.

Creates an empty ACL message object.

Definition at line 114 of file MCAclMessage.cs.

Referenced by LibMC.MCAclMessage.Reply().

### 7.1.4 Member Function Documentation

#### 7.1.4.1 void LibMC.MCAclMessage.New ()

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.

#### 7.1.4.2 MCAclMessage LibMC.MCAclMessage.Reply (MCAclMessage *acl\_message*)

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 LibMC.MCAclMessage.AclMsg, and LibMC.MCAclMessage.MCAclMessage().

#### 7.1.4.3 int LibMC.MCAclMessage.SetPerformative (MC\_FipaPerformative\_e *performative*)

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.

**7.1.4.4 int LibMC.MCAclMessage.SetSender (String name, String address)**

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.

**7.1.4.5 int LibMC.MCAclMessage.AddReceiver (String name, String address)**

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.

**7.1.4.6 int LibMC.MCAclMessage.AddReplyTo (String *name*, String *address*)**

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.

**7.1.4.7 int LibMC.MCAclMessage.SetContent (String *content*)**

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.

**7.1.4.8 int LibMC.MCAclMessage.Destroy ()**

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.

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

- </home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/MCAclMessage.cs>

## 7.2 LibMC.MCAgency Class Reference

Wrapper class for MCAgency\_t structure.

### Public Types

- enum [MCAgencyState](#)  
*Enum for describing the state of the agency.*
- enum [ChShellType](#)  
*Ch shell type.*
- enum [MC\\_ThreadIndex\\_e](#)  
*Enum for describing the different threads that Mobile-C uses.*
- enum [MC\\_SteerCommand\\_e](#)  
*Available commands for MC\_Steer.*

### 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 [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](#) (int signals)  
*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

- int [Port](#) [get, set]  
*Accessor for the port number of the agency.*
- [MCAgencyState State](#) [get]  
*Accessor for the agency state.*

### 7.2.1 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 329 of file MCAgency.cs.

### 7.2.2 Member Enumeration Documentation

#### 7.2.2.1 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 337 of file MCAgency.cs.

#### 7.2.2.2 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 351 of file MCAgency.cs.

#### 7.2.2.3 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 Managment system

*MC\_THREAD\_ACC* Agency communications

*MC\_THREAD\_CP* Command Prompt

*MC\_THREAD\_AGENT* Agent threads

Definition at line 79 of file MCEExports.cs.

#### 7.2.2.4 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.

### 7.2.3 Constructor & Destructor Documentation

#### 7.2.3.1 LibMC.MCAgency.MCAgency ()

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 369 of file MCAgency.cs.

### 7.2.4 Member Function Documentation

#### 7.2.4.1 int LibMC.MCAgency.Initialize ()

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 445 of file MCAgency.cs.

#### 7.2.4.2 `int LibMC.MCAgency.End ()`

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 467 of file MCAgency.cs.

#### 7.2.4.3 `int LibMC.MCAgency.ChInitializeOptions (ChShellType shellType, String home)`

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\_REGULARCH 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 485 of file MCAgency.cs.

#### 7.2.4.4 `int LibMC.MCAgency.SetThreadsAllOff ()`

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 516 of file MCAgency.cs.

**7.2.4.5 int LibMC.MCAgency.SetThreadOn (MC\_ThreadIndex\_e *index*)**

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 532 of file MCAgency.cs.

**7.2.4.6 int LibMC.MCAgency.SetThreadOff (MC\_ThreadIndex\_e *index*)**

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 547 of file MCAgency.cs.

**7.2.4.7 int LibMC.MCAgency.HaltAgency ()**

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 562 of file MCAgency.cs.

**7.2.4.8 int LibMC.MCAgency.ResumeAgency ()**

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 579 of file MCAgency.cs.

**7.2.4.9 int LibMC.MCAgency.SetDefaultAgentStatus (MCAgent.MC\_AgentStatus\_e status)**

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 595 of file MCAgency.cs.

**7.2.4.10 MCAgent LibMC.MCAgency.WaitRetrieveAgent ()**

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 608 of file MCAgency.cs.

**7.2.4.11 int LibMC.MCAgency.WaitAgent ()**

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 625 of file MCAgency.cs.

**7.2.4.12 int LibMC.MCAgency.SendAgentMigrationMessageFile (String *filename*, String *hostname*, int *port*)**

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 644 of file MCAgency.cs.

**7.2.4.13 int LibMC.MCAgency.LoadAgentMigrationMessageFile (String *filename*)**

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 658 of file MCAgency.cs.

References LibMC.MCAgency.Port.

**7.2.4.14 int LibMC.MCAgency.SendAgentMigrationMessage (String *message*, String *hostname*, int *port*)**

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 673 of file MCAgency.cs.

**7.2.4.15 int LibMC.MCAgency.CondBroadcast (int *id*)**

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 691 of file MCAgency.cs.

**7.2.4.16 int LibMC.MCAgency.CondSignal (int *id*)**

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 705 of file MCAgency.cs.

**7.2.4.17 int LibMC.MCAgency.CondReset (int *id*)**

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 720 of file MCAgency.cs.

**7.2.4.18 int LibMC.MCAgency.CondWait (int *id*)**

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 735 of file MCAgency.cs.

**7.2.4.19 int LibMC.MCAgency.MutexLock (int *id*)**

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 750 of file MCAgency.cs.

**7.2.4.20 int LibMC.MCAgency.MutexUnlock (int *id*)**

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 764 of file MCAgency.cs.

**7.2.4.21 int LibMC.MCAgency.SemaphorePost (int *id*)**

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 778 of file MCAgency.cs.

#### 7.2.4.22 **int LibMC.MCAgency.SemaphoreWait (int *id*)**

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 793 of file MCAgency.cs.

#### 7.2.4.23 **int LibMC.MCAgency.ResetSignal ()**

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 806 of file MCAgency.cs.

#### 7.2.4.24 **int LibMC.MCAgency.SyncDelete (int *id*)**

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 820 of file MCAgency.cs.

#### 7.2.4.25 **int LibMC.MCAgency.SyncInit (int *id*)**

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 836 of file MCAgency.cs.

**7.2.4.26 int LibMC.MCAgency.WaitSignal (int *signals*)**

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 849 of file MCAgency.cs.

**7.2.4.27 int LibMC.MCAgency.BarrierDelete (int *id*)**

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 863 of file MCAgency.cs.

**7.2.4.28 int LibMC.MCAgency.BarrierInit (int *id*, int *num\_procs*)**

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 877 of file MCAgency.cs.

**7.2.4.29 MC\_SteerCommand\_e LibMC.MCAgency.SteerControl ()**

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 895 of file MCAgency.cs.

**7.2.4.30 int LibMC.MCAgency.Steer (IntPtr funcptr, IntPtr arg)**

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 913 of file MCAgency.cs.

**7.2.4.31 int LibMC.MCAgency.RegisterService (MCAgent agent, int agentID, String agentName, String[] serviceNames, int numServices)**

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 938 of file MCAgency.cs.

References LibMC.MCAgent.Agent.

**7.2.4.32 int LibMC.MCAgency.SearchForService (String searchString, IntPtr agentNames, IntPtr serviceNames, IntPtr agentIDs, IntPtr numResults)**

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 961 of file MCAgency.cs.

**7.2.4.33 int LibMC.MCAgency.AddAgent (MCAgent agent)**

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 979 of file MCAgency.cs.

References LibMC.MCAgent.Agent.

**7.2.4.34 MCAgent LibMC.MCAgency.FindAgentByName (String *name*)**

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 992 of file MCAgency.cs.

**7.2.4.35 MCAgent LibMC.MCAgency.FindAgentByID (int *id*)**

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 1005 of file MCAgency.cs.

**7.2.4.36 MCAgent LibMC.MCAgency.RetrieveAgent ()**

Retrieve an agent from the agency.

Really not sure.

**Returns:**

The return value of the underlying MC\_CondBroadcast function.

Definition at line 1017 of file MCAgency.cs.

**7.2.4.37 int LibMC.MCAgency.AclSend (MCAclMessage *acl\_message*)**

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 1034 of file MCAgency.cs.

References LibMC.MCAclMessage.AclMsg.

**7.2.4.38 int LibMC.MCAgency.MainLoop ()**

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 1047 of file MCAgency.cs.

**7.2.5 Property Documentation****7.2.5.1 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 404 of file MCAgency.cs.

Referenced by LibMC.MCAgency.LoadAgentMigrationMessageFile().

**7.2.5.2 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 424 of file MCAgency.cs.

The documentation for this class was generated from the following files:

- [/home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/MCAgency.cs](#)
- [/home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/MCExports.cs](#)



## 7.3 LibMC.MCAgent Class Reference

Wrapper class for MCAgent\_t structure.

### Public Types

- enum [MC\\_AgentType\\_e](#)  
*Enum for describing the type of an agent.*
- enum [MC\\_AgentStatus\\_e](#)  
*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.*
- 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.*

## Properties

- 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.*

### 7.3.1 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 60 of file MCAgent.cs.

### 7.3.2 Member Enumeration Documentation

#### 7.3.2.1 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 74 of file MCAgent.cs.

### 7.3.2.2 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 87 of file MCAgent.cs.

## 7.3.3 Constructor & Destructor Documentation

### 7.3.3.1 LibMC.MCAgent.MCAgent ()

Default constructor.

Creates an empty agent.

Definition at line 103 of file MCAgent.cs.

## 7.3.4 Member Function Documentation

### 7.3.4.1 override string LibMC.MCAgent.ToString ()

Display the agent's fields.

Formats an returns a string with all of the agents 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 125 of file MCAgent.cs.

References LibMC.MCAgent.AgentID, LibMC.MCAgent.AgentName,  
LibMC.MCAgent.AgentNumTasks, LibMC.MCAgent.AgentStatus, and LibMC.MCAgent.AgentType.

**7.3.4.2 int LibMC.MCAgent.DeleteAgent ()**

Deletes an agent.

Deletes an agent from the agency.

**Returns:**

The return value of the underlying MC\_DeleteAgent function call.

Definition at line 299 of file MCAgent.cs.

**7.3.4.3 String LibMC.MCAgent.GetAgentXMLString ()**

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 312 of file MCAgent.cs.

**7.3.4.4 int LibMC.MCAgent.PrintAgentCode ()**

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 325 of file MCAgent.cs.

**7.3.4.5 String LibMC.MCAgent.RetrieveAgentCode ()**

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 337 of file MCAgent.cs.

**7.3.4.6 int LibMC.MCAgent.TerminateAgent ()**

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 350 of file MCAgent.cs.

**7.3.4.7 int LibMC.MCAgent.AclPost (MCAclMessage *message*)**

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 375 of file MCAgent.cs.

References LibMC.MCAclMessage.AclMsg.

**7.3.4.8 MCAclMessage LibMC.MCAgent.AclRetrieve ()**

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 391 of file MCAgent.cs.

**7.3.4.9 MCAclMessage LibMC.MCAgent.AclWaitRetrieve ()**

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 410 of file MCAgent.cs.

**7.3.4.10 int LibMC.MCAgent.CallAgentFunc (String *funcName*, IntPtr *retval*, IntPtr *varg*)**

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!

**Todo**

Find a better way to handle retval and varg

Definition at line 434 of file MCAgent.cs.

**7.3.4.11 IntPtr LibMC.MCAgent.GetAgentExecEngine ()**

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 451 of file MCAgent.cs.

**7.3.4.12 int LibMC.MCAgent.GetAgentReturnData (int *task\_num*, IntPtr *data*, IntPtr *dim*, IntPtr *extent*)**

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 472 of file MCAgent.cs.

### 7.3.5 Property Documentation

#### 7.3.5.1 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 180 of file MCAgent.cs.

Referenced by LibMC.MCAgent.ToString().

#### 7.3.5.2 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 200 of file MCAgent.cs.

Referenced by LibMC.MCAgent.ToString().

#### 7.3.5.3 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 219 of file MCAgent.cs.

Referenced by LibMC.MCAgent.ToString().

**7.3.5.4 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 241 of file MCAgent.cs.

Referenced by LibMC.MCAgent.ToString().

**7.3.5.5 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 263 of file MCAgent.cs.

Referenced by LibMC.MCAgent.ToString().

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

- </home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/MCAgent.cs>



## Chapter 8

# LibMC.NET File Documentation

### 8.1 `/home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/MCAclMess` File Reference

#### Namespaces

- namespace [LibMC](#)
- namespace **System**
- namespace **System.Collections.Generic**
- namespace **System.Text**

#### Classes

- class [LibMC.MCAclMessage](#)  
*Encapsulates ACL messages in the Mobile-C library.*

#### 8.1.1 Detailed Description

Defines the MCAclMessage object and its member functions.

Definition in file [MCAclMessage.cs](#).

## 8.2 /home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/MCAgency.c File Reference

### Namespaces

- namespace [LibMC](#)
- namespace **System.Runtime.InteropServices**

### Classes

- class [LibMC.MCAgency](#)  
*Wrapper class for MCAgency\_t structure.*

#### 8.2.1 Detailed Description

Defines the MCAgency object and its member functions.

Definition in file [MCAgency.cs](#).

## 8.3 /home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/MCAgent.cs File Reference

### Namespaces

- namespace [LibMC](#)

### Classes

- class [LibMC.MCAgent](#)  
*Wrapper class for MCAgent\_t structure.*

#### 8.3.1 Detailed Description

Defines the MCAgent object and its member functions.

Definition in file [MCAgent.cs](#).

## 8.4 /home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/MCExports. File Reference

### Namespaces

- namespace [LibMC](#)

### Classes

- class [LibMC.MCAgency](#)  
*Wrapper class for MCAgency\_t structure.*

#### 8.4.1 Detailed Description

Imports functions, structs, and enums from the Mobile-C library.

Definition in file [MCExports.cs](#).

## **Chapter 9**

# **LibMC.NET Example Documentation**

### **9.1 LibMCConsole/Program.cs**

Basic Mobile-C console demo program

## 9.2 LibMCCppEx/LibMCCppEx.cpp

Demonstrates using LibMC.NET from a VC++ program.

## **9.3 LibMCFipaTest/Program.cs**

Mobile-C FIPA ACL message demo program.

## 9.4 LibMCGui/Form1.cs

Basic Mobile-C Windows Forms demo program



## **9.5 LibMCMiscTest/Program.cs**

Demonstrates miscellaneous Mobile-C functions.

## 9.6 LibMCVbEx/Form1.vb

Demonstrates using LibMC.NET from a VB program.

## Chapter 10

# LibMC.NET Page Documentation

### 10.1 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.

## 10.2 Installing LibMC.NET

Installing LibMC.NET is straightforward but involves several steps.

### 10.2.1 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/>.

### 10.2.2 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](#).

### 10.2.3 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.

### 10.2.4 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.

### 10.2.5 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.

## 10.3 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.

### 10.3.1 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.

## 10.4 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.

### 10.4.1 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.

### 10.4.2 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.

### 10.4.3 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.

## 10.5 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.



## 10.6 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();
```

## 10.7 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());
```

## 10.8 Todo List

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

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

Member **LibMC::MCAgency.RegisterService(MCAgent agent, int agentID, String agentName, String[] serviceNames, IntPtr arg)** Test MC\_RegisterService and MC\_SearchForService.

Member **LibMC::MCAgency.SearchForService(String searchString, IntPtr agentNames, IntPtr serviceNames, IntPtr arg)** Implement SearchForService

Member **LibMC::MCAgent.CallAgentFunc(String funcName, IntPtr retval, IntPtr varg)** Find a better way to handle retval and varg

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

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

## 10.9 Bug List

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

Member `LibMC::MCAgency.SearchForService(String searchString, IntPtr agentNames, IntPtr serviceNames, IntPtr a`  
MC\_SearchForService is not yet implemented.

# Index

/home/dko/projects/mobilec/trunk/src/win32/LibMC.nclibmc/MCAclMessage.cs,  
45  
LibMC::MCAgency, 24  
/home/dko/projects/mobilec/trunk/src/win32/LibMC.nclibmc/MCAgency.cs,  
46  
LibMC::MCAgency, 22  
/home/dko/projects/mobilec/trunk/src/win32/LibMC.nclibmc/MCAgent.cs,  
47  
LibMC::MCAgency, 27  
/home/dko/projects/mobilec/trunk/src/win32/LibMC.nclibmc/MCExports.cs,  
48  
LibMC::MCAgency, 28  
CondSignal  
LibMC::MCAgency, 28  
CondWait  
LibMC::MCAgency, 28  
DeleteAgent  
LibMC::MCAgent, 39  
Destroy  
LibMC::MCAclMessage, 17  
End  
LibMC::MCAgency, 23  
Ended  
LibMC::MCAgency, 22  
FindAgentByID  
LibMC::MCAgency, 34  
FindAgentByName  
LibMC::MCAgency, 34  
FIPA\_ACCEPT\_PROPOSAL  
LibMC::MCAclMessage, 14  
FIPA\_AGREE  
LibMC::MCAclMessage, 14  
FIPA\_CALL\_FOR\_PROPOSAL  
LibMC::MCAclMessage, 14  
FIPA\_CANCEL  
LibMC::MCAclMessage, 14  
FIPA\_CONFIRM  
LibMC::MCAclMessage, 14  
FIPA\_DISCONFIRM  
LibMC::MCAclMessage, 14  
FIPA\_ERROR  
LibMC::MCAclMessage, 14  
FIPA\_FAILURE  
LibMC::MCAclMessage, 14  
FIPA\_INFORM  
LibMC::MCAclMessage, 14  
/home/dko/projects/mobilec/trunk/src/win32/LibMC.nclibmc/MCAclMessage.cs,  
45  
LibMC::MCAgency, 24  
/home/dko/projects/mobilec/trunk/src/win32/LibMC.nclibmc/MCAgency.cs,  
46  
LibMC::MCAgency, 22  
/home/dko/projects/mobilec/trunk/src/win32/LibMC.nclibmc/MCAgent.cs,  
47  
LibMC::MCAgency, 27  
/home/dko/projects/mobilec/trunk/src/win32/LibMC.nclibmc/MCExports.cs,  
48  
LibMC::MCAgency, 28  
CondSignal  
LibMC::MCAgency, 28  
CondWait  
LibMC::MCAgency, 28  
DeleteAgent  
LibMC::MCAgent, 39  
Destroy  
LibMC::MCAclMessage, 17  
End  
LibMC::MCAgency, 23  
Ended  
LibMC::MCAgency, 22  
FindAgentByID  
LibMC::MCAgency, 34  
FindAgentByName  
LibMC::MCAgency, 34  
FIPA\_ACCEPT\_PROPOSAL  
LibMC::MCAclMessage, 14  
FIPA\_AGREE  
LibMC::MCAclMessage, 14  
FIPA\_CALL\_FOR\_PROPOSAL  
LibMC::MCAclMessage, 14  
FIPA\_CANCEL  
LibMC::MCAclMessage, 14  
FIPA\_CONFIRM  
LibMC::MCAclMessage, 14  
FIPA\_DISCONFIRM  
LibMC::MCAclMessage, 14  
FIPA\_ERROR  
LibMC::MCAclMessage, 14  
FIPA\_FAILURE  
LibMC::MCAclMessage, 14  
FIPA\_INFORM  
LibMC::MCAclMessage, 14  
AclPost  
LibMC::MCAgent, 41  
AclRetrieve  
LibMC::MCAgent, 41  
AclSend  
LibMC::MCAgency, 34  
AclWaitRetrieve  
LibMC::MCAgent, 41  
AddAgent  
LibMC::MCAgency, 33  
AddReceiver  
LibMC::MCAclMessage, 16  
AddReplyTo  
LibMC::MCAclMessage, 16  
AgentID  
LibMC::MCAgent, 43  
AgentName  
LibMC::MCAgent, 43  
AgentNumTasks  
LibMC::MCAgent, 43  
AgentStatus  
LibMC::MCAgent, 44  
AgentType  
LibMC::MCAgent, 44  
BarrierDelete  
LibMC::MCAgency, 31  
BarrierInit  
LibMC::MCAgency, 31  
CallAgentFunc  
LibMC::MCAgent, 41  
CH\_REGULARCH  
LibMC::MCAgency, 22  
CH\_SAFECH  
LibMC::MCAgency, 22

- FIPA\_INFORM\_IF
  - LibMC::MCAclMessage, 14
- FIPA\_INFORM\_REF
  - LibMC::MCAclMessage, 14
- FIPA\_NOT\_UNDERSTOOD
  - LibMC::MCAclMessage, 14
- FIPA\_PROPOGATE
  - LibMC::MCAclMessage, 14
- FIPA\_PROPOSE
  - LibMC::MCAclMessage, 14
- FIPA\_PROXY
  - LibMC::MCAclMessage, 14
- FIPA\_QUERY\_IF
  - LibMC::MCAclMessage, 14
- FIPA\_QUERY\_REF
  - LibMC::MCAclMessage, 14
- FIPA\_REFUSE
  - LibMC::MCAclMessage, 14
- FIPA\_REJECT\_PROPOSAL
  - LibMC::MCAclMessage, 14
- FIPA\_REQUEST
  - LibMC::MCAclMessage, 15
- FIPA\_REQUEST\_WHEN
  - LibMC::MCAclMessage, 15
- FIPA\_REQUEST\_WHENEVER
  - LibMC::MCAclMessage, 15
- FIPA\_SUBSCRIBE
  - LibMC::MCAclMessage, 15
- FIPA\_ZERO
  - LibMC::MCAclMessage, 14
- GetAgentExecEngine
  - LibMC::MCAgent, 42
- GetAgentReturnData
  - LibMC::MCAgent, 42
- GetAgentXMLString
  - LibMC::MCAgent, 40
- HaltAgency
  - LibMC::MCAgency, 25
- Halted
  - LibMC::MCAgency, 22
- Initialize
  - LibMC::MCAgency, 23
- Initialized
  - LibMC::MCAgency, 22
- LibMC, 11
- LibMC.Properties, 12
- LibMC::MCAclMessage, 13
  - AddReceiver, 16
  - AddReplyTo, 16
  - Destroy, 17
  - FIPA\_ACCEPT\_PROPOSAL, 14
  - FIPA\_AGREE, 14
  - FIPA\_CALL\_FOR\_PROPOSAL, 14
  - FIPA\_CANCEL, 14
  - FIPA\_CONFIRM, 14
  - FIPA\_DISCONFIRM, 14
  - FIPA\_ERROR, 14
  - FIPA\_FAILURE, 14
  - FIPA\_INFORM, 14
  - FIPA\_INFORM\_IF, 14
  - FIPA\_INFORM\_REF, 14
  - FIPA\_NOT\_UNDERSTOOD, 14
  - FIPA\_PROPOGATE, 14
  - FIPA\_PROPOSE, 14
  - FIPA\_PROXY, 14
  - FIPA\_QUERY\_IF, 14
  - FIPA\_QUERY\_REF, 14
  - FIPA\_REFUSE, 14
  - FIPA\_REJECT\_PROPOSAL, 14
  - FIPA\_REQUEST, 15
  - FIPA\_REQUEST\_WHEN, 15
  - FIPA\_REQUEST\_WHENEVER, 15
  - FIPA\_SUBSCRIBE, 15
  - FIPA\_ZERO, 14
  - MC\_FipaPerformative\_e, 14
  - MCAclMessage, 15
  - New, 15
  - Reply, 15
  - SetContent, 17
  - SetPerformative, 15
  - SetSender, 16
- LibMC::MCAgency, 19
  - AclSend, 34
  - AddAgent, 33
  - BarrierDelete, 31
  - BarrierInit, 31
  - CH\_REGULARCH, 22
  - CH\_SAFECH, 22
  - ChInitializeOptions, 24
  - ChShellType, 22
  - CondBroadcast, 27
  - CondReset, 28
  - CondSignal, 28
  - CondWait, 28
  - End, 23
  - Ended, 22
  - FindAgentByID, 34
  - FindAgentByName, 34
  - HaltAgency, 25
  - Halted, 22
  - Initialize, 23
  - Initialized, 22
  - LoadAgentMigrationMessageFile, 27
  - MainLoop, 35

- MC\_RESTART, 23
- MC\_RUN, 23
- MC\_STOP, 23
- MC\_SUSPEND, 23
- MC\_THREAD\_ACC, 23
- MC\_THREAD\_AGENT, 23
- MC\_THREAD\_AMS, 22
- MC\_THREAD\_CP, 23
- MC\_THREAD\_DF, 22
- MC\_SterCommand\_e, 23
- MC\_ThreadIndex\_e, 22
- MCAgency, 23
- MCAgencyState, 22
- MutexLock, 29
- MutexUnlock, 29
- NoState, 22
- Port, 35
- RegisterService, 32
- ResetSignal, 30
- ResumeAgency, 25
- RetrieveAgent, 34
- Running, 22
- SearchForService, 33
- SemaphorePost, 29
- SemaphoreWait, 29
- SendAgentMigrationMessage, 27
- SendAgentMigrationMessageFile, 26
- SetDefaultAgentStatus, 26
- SetThreadOff, 25
- SetThreadOn, 24
- SetThreadsAllOff, 24
- State, 35
- Steer, 32
- SteerControl, 31
- SyncDelete, 30
- SyncInit, 30
- WaitAgent, 26
- WaitRetrieveAgent, 26
- WaitSignal, 31
- LibMC::MCAgent, 37
  - AclPost, 41
  - AclRetrieve, 41
  - AclWaitRetrieve, 41
  - AgentID, 43
  - AgentName, 43
  - AgentNumTasks, 43
  - AgentStatus, 44
  - AgentType, 44
  - CallAgentFunc, 41
  - DeleteAgent, 39
  - GetAgentExecEngine, 42
  - GetAgentReturnData, 42
  - GetAgentXMLString, 40
  - MC\_AGENT\_ACTIVE, 39
  - MC\_AGENT\_NEUTRAL, 39
  - MC\_AGENT\_SUSPENDED, 39
  - MC\_LOCAL\_AGENT, 38
  - MC\_NO\_STATUS, 39
  - MC\_NONE, 38
  - MC\_REMOTE\_AGENT, 38
  - MC\_RETURN\_AGENT, 38
  - MC\_WAIT\_CH, 39
  - MC\_WAIT\_FINISHED, 39
  - MC\_WAIT\_MESSGSEND, 39
  - MC\_AgentStatus\_e, 38
  - MC\_AgentType\_e, 38
  - MCAgent, 39
  - PrintAgentCode, 40
  - RetrieveAgentCode, 40
  - TerminateAgent, 40
  - ToString, 39
- LoadAgentMigrationMessageFile
  - LibMC::MCAgency, 27
- MainLoop
  - LibMC::MCAgency, 35
- MC\_AGENT\_ACTIVE
  - LibMC::MCAgent, 39
- MC\_AGENT\_NEUTRAL
  - LibMC::MCAgent, 39
- MC\_AGENT\_SUSPENDED
  - LibMC::MCAgent, 39
- MC\_LOCAL\_AGENT
  - LibMC::MCAgent, 38
- MC\_NO\_STATUS
  - LibMC::MCAgent, 39
- MC\_NONE
  - LibMC::MCAgent, 38
- MC\_REMOTE\_AGENT
  - LibMC::MCAgent, 38
- MC\_RESTART
  - LibMC::MCAgency, 23
- MC\_RETURN\_AGENT
  - LibMC::MCAgent, 38
- MC\_RUN
  - LibMC::MCAgency, 23
- MC\_STOP
  - LibMC::MCAgency, 23
- MC\_SUSPEND
  - LibMC::MCAgency, 23
- MC\_THREAD\_ACC
  - LibMC::MCAgency, 23
- MC\_THREAD\_AGENT
  - LibMC::MCAgency, 23
- MC\_THREAD\_AMS
  - LibMC::MCAgency, 22
- MC\_THREAD\_CP
  - LibMC::MCAgency, 23



- MC\_THREAD\_DF
  - LibMC::MCAgent, 22
- MC\_WAIT\_CH
  - LibMC::MCAgent, 39
- MC\_WAIT\_FINISHED
  - LibMC::MCAgent, 39
- MC\_WAIT\_MESSGSEND
  - LibMC::MCAgent, 39
- MC\_AgentStatus\_e
  - LibMC::MCAgent, 38
- MC\_AgentType\_e
  - LibMC::MCAgent, 38
- MC\_FipaPerformative\_e
  - LibMC::MCAclMessage, 14
- MC\_SteerCommand\_e
  - LibMC::MCAgent, 23
- MC\_ThreadIndex\_e
  - LibMC::MCAgent, 22
- MCAclMessage
  - LibMC::MCAclMessage, 15
- MCAgency
  - LibMC::MCAgent, 23
- MCAgencyState
  - LibMC::MCAgent, 22
- MCAgent
  - LibMC::MCAgent, 39
- MutexLock
  - LibMC::MCAgent, 29
- MutexUnlock
  - LibMC::MCAgent, 29
- New
  - LibMC::MCAclMessage, 15
- NoState
  - LibMC::MCAgent, 22
- Port
  - LibMC::MCAgent, 35
- PrintAgentCode
  - LibMC::MCAgent, 40
- RegisterService
  - LibMC::MCAgent, 32
- Reply
  - LibMC::MCAclMessage, 15
- ResetSignal
  - LibMC::MCAgent, 30
- ResumeAgency
  - LibMC::MCAgent, 25
- RetrieveAgent
  - LibMC::MCAgent, 34
- RetrieveAgentCode
  - LibMC::MCAgent, 40
- Running
  - LibMC::MCAgent, 22
- SearchForService
  - LibMC::MCAgent, 33
- SemaphorePost
  - LibMC::MCAgent, 29
- SemaphoreWait
  - LibMC::MCAgent, 29
- SendAgentMigrationMessage
  - LibMC::MCAgent, 27
- SendAgentMigrationMessageFile
  - LibMC::MCAgent, 26
- SetContent
  - LibMC::MCAclMessage, 17
- SetDefaultAgentStatus
  - LibMC::MCAgent, 26
- SetPerformative
  - LibMC::MCAclMessage, 15
- SetSender
  - LibMC::MCAclMessage, 16
- SetThreadOff
  - LibMC::MCAgent, 25
- SetThreadOn
  - LibMC::MCAgent, 24
- SetThreadsAllOff
  - LibMC::MCAgent, 24
- State
  - LibMC::MCAgent, 35
- Steer
  - LibMC::MCAgent, 32
- SteerControl
  - LibMC::MCAgent, 31
- SyncDelete
  - LibMC::MCAgent, 30
- SyncInit
  - LibMC::MCAgent, 30
- TerminateAgent
  - LibMC::MCAgent, 40
- ToString
  - LibMC::MCAgent, 39
- WaitAgent
  - LibMC::MCAgent, 26
- WaitRetrieveAgent
  - LibMC::MCAgent, 26
- WaitSignal
  - LibMC::MCAgent, 31