



SCE-RT SDK Medius Game Communication Library (MGCL) Release 2.10

Reference

May 2005

© 2005 Sony Computer Entertainment Inc.

All rights reserved.

Publication date: May 2005

Sony Computer Entertainment Inc.
2-6-21, Minami-Aoyama, Minato-ku
Tokyo 107-0062, Japan

Sony Computer Entertainment America
919 E. Hillsdale Blvd.
Foster City, CA 94404, U.S.A

Sony Computer Entertainment Europe
30 Golden Square
London W1F 9LD, U.K.

The *SCE-RT SDK Medius Game Communication Library (MGCL) Release 2.10 – Reference* is supplied pursuant to and subject to the terms of the Sony Computer Entertainment PlayStation® license agreements.

The *SCE-RT SDK Medius Game Communication Library (MGCL) Release 2.10 – Reference* is intended for distribution to and use by only Sony Computer Entertainment licensed Developers and Publishers in accordance with the PlayStation® license agreements.

Unauthorized reproduction, distribution, lending, rental or disclosure to any third party, in whole or in part, of this document is expressly prohibited by law and by the terms of the Sony Computer Entertainment PlayStation® license agreements.

Ownership of the physical property of the document is retained by and reserved by Sony Computer Entertainment. Alteration to or deletion, in whole or in part, of the document, its presentation, or its contents is prohibited.

The information in the *SCE-RT SDK Medius Game Communication Library (MGCL) Release 2.10 – Reference* is subject to change without notice. The content of this document is Confidential Information of Sony Computer Entertainment.

 and PlayStation are registered trademarks of Sony Computer Entertainment Inc. All other trademarks are property of their respective owners and/or their licensors.

Table of Contents

About This Manual	vii
Changes Since Last Release	vii
Related Documentation	vii
Manual Structure	vii
Developer Reference Series	vii
Typographic Conventions	viii
Developer Support	viii
Chapter 1: Defines/Macros	1-1
MAX_CLIENTS_PER_WORLD	1-3
MEDIUS_GAME_COMM_LIBRARY_VERSION_NUMBER_MAXLEN	1-4
MGCL_ACCESSKEY_MAXLEN	1-5
MGCL_GAMENAME_MAXLEN	1-6
MGCL_GAMEPASSWORD_MAXLEN	1-7
MGCL_GAMESTATS_MAXLEN	1-8
MGCL_MESSAGEID_MAXLEN	1-9
MGCL_SERVERIP_MAXLEN	1-10
MGCL_SERVERPORT_MAXLEN	1-11
MGCL_SERVERVERSION_MAXLEN	1-12
MGCL_SESSIONKEY_MAXLEN	1-13
MGCL_VERSION_BUILD	1-14
MGCL_VERSION_MAJOR	1-15
MGCL_VERSION_MINOR	1-16
Chapter 2: Enumerated Types	2-1
DMEInitStatus	2-3
MGCL_ALERT_LEVEL	2-4
MGCL_ERROR_CODE	2-5
MGCL_EVENT_TYPE	2-7
MGCL_GAME_HOST_TYPE	2-8
MGCL_SERVER_ATTRIBUTES	2-9
MGCL_TRUST_LEVEL	2-10
MGCL_WORLD_ATTRIBUTES	2-11
MGCL_WORLD_STATUS	2-12
Chapter 3: Structures	3-1
MediusServerAuthenticationRequest	3-3
MediusServerAuthenticationResponse	3-4
MediusServerConnectGamesRequest	3-5
MediusServerConnectGamesResponse	3-6
MediusServerConnectNotificationType	3-7
MediusServerCreateGameOnMeRequest	3-8
MediusServerCreateGameOnMeResponse	3-10
MediusServerCreateGameRequest	3-11
MediusServerCreateGameResponse	3-12
MediusServerCreateGameWithAttributesRequest	3-13
MediusServerCreateGameWithAttributesResponse	3-14
MediusServerDisconnectPlayerRequest	3-15
MediusServerEndGameOnMeRequest	3-16
MediusServerEndGameOnMeResponse	3-17
MediusServerEndGameRequest	3-18

MediusServerEndGameResponse	3-19
MediusServerJoinGameRequest	3-20
MediusServerJoinGameResponse	3-21
MediusServerMoveGameWorldOnMeRequest	3-22
MediusServerMoveGameWorldOnMeResponse	3-23
MediusServerReportType	3-24
MediusServerSessionBeginRequest	3-25
MediusServerSessionBeginResponse	3-26
MediusServerSessionEndRequest	3-27
MediusServerSessionEndResponse	3-28
MediusServerSetAttributesRequest	3-29
MediusServerSetAttributesResponse	3-30
MediusServerWorldReportOnMe	3-31
MediusServerWorldStatusRequest	3-33
MediusServerWorldStatusResponse	3-34
MGCLConnectInParams	3-35
MGCLConnectOutParams	3-36
MGCLCreateGameOnMeRequestInParams	3-37
MGCLDisconnectParams	3-38
MGCLInitializeInParams	3-39
MGCLInitializeOutParams	3-41
Chapter 4: Callback Functions	4-1
MGCLConnectCallback	4-3
MGCLServerAuthenticationCallback	4-4
MGCLServerConnectGamesCallback	4-5
MGCLServerCreateGameCallback	4-6
MGCLServerCreateGameOnMeCallback	4-7
MGCLServerCreateGameWithAttributesCallback	4-8
MGCLServerDisconnectPlayerCallback	4-9
MGCLServerEndGameCallback	4-10
MGCLServerEndGameOnMeCallback	4-11
MGCLServerJoinGameCallback	4-12
MGCLServerMoveGameWorldOnMeCallback	4-13
MGCLServerSessionBeginCallback	4-14
MGCLServerSessionEndCallback	4-15
MGCLServerSetAttributesCallback	4-16
MGCLServerWorldStatusCallback	4-17
Chapter 5: Functions	5-1
MGCLAuthenticationSend	5-3
MGCLClose	5-4
MGCLConnect	5-5
MGCLConnectGamesResponse	5-6
MGCLCreateGameOnMeRequest	5-7
MGCLCreateGameResponse	5-8
MGCLCreateGameWithAttributesResponse	5-9
MGCLDisconnect	5-10
MGCLEndGameOnMeRequest	5-11
MGCLEndGameResponse	5-12
MGCLGetMGCLBuildTimeStamp	5-13
MGCLGetMGCLVersion	5-14
MGCLGetServerTimestamp	5-15

MGCLInitialize	5-16
MGCLIPStringToBinary	5-17
MGCLJoinGameResponse	5-18
MGCLMoveGameWorldOnMe	5-19
MGCLSendConnectNotification	5-20
MGCLSendServerReport	5-21
MGCLSendWorldReportOnMe	5-22
MGCLSessionBegin	5-23
MGCLSessionEnd	5-24
MGCLSetAuthConnectParams	5-25
MGCLSetDefaultConnectParams	5-26
MGCLSetDefaultDisconnectParams	5-27
MGCLSetServerAttributes	5-28
MGCLUpdate	5-29
MGCLWorldStatusResponse	5-30
Index	I-1

This page intentionally left blank.

About This Manual

The *SCE-RT SDK Medius Game Communication Library (MGCL) Release 2.10 – Reference* provides a description of the various functionalities of the Medius Game Communication Library (MGCL). This API is an add-on to the DME API and Medius API provided by Sony Computer Entertainment America (SCEA).

Please forward any questions about this document to scert-support@scea.com.

Changes Since Last Release

`MGCLCreateGameOnMe()` nows takes in a HDME input parameter (needed if `HostType == PeerToPeer`).

Related Documentation

Related documentation for the SCE-RT SDK Medius Game Communication Library (MGCL) Release 2.10 – Reference consists of the following:

SCE-RT DME API Reference

SCE-RT Medius API Reference

You should read this manual in conjunction with:

SCE-RT MGCL Overview

SCE-RT DME Overview

SCE-RT Medius Overview

Note: the Developer Support Websites (<https://www.ps2-pro.com/> and <https://psp.scedev.net>) post current developments regarding the Network Gaming Service and also provides notice of future documentation releases and upgrades.

Manual Structure

Section	Description
Ch. 1: Defines/Macros	Provides Defines/Macros for the MGCL
Ch. 2: Enumerated Types	Provides Enumerated Types for the MGCL
Ch. 3: Structures	Provides Structures for the MGCL
Ch. 4: Callback Functions	Provides Callback Functions for the MGCL
Ch. 5: Functions	Provides Functions for the MGCL
Index	Provides an Index for the MGCL

Developer Reference Series

This manual is part of the *Developer Reference Series*, a series of technical reference volumes covering all aspects of PlayStation® development. The complete series is listed below:

Manual	Description
<code>SCE-RT_SDK_DME_API_Overview</code>	Distributed Memory Engine (DME) Overview. Used for in-game networked data management.
<code>SCE-RT_SDK_DME_API_Reference</code>	API for the DME
<code>SCE-RT_SDK_MEDIUS_API_Overview</code>	Medius client API overview. Used for user authentication, lobby chat, and player matching functionality.
<code>SCE-RT_SDK_MEDIUS_API_Reference</code>	API for the Medius client.

Manual	Description
SCE-RT_SDK_MGCL_API_Overview	Medius Game Communication Library (MGCL) Overview. Used for game hosting and peer-to-peer play.
SCE-RT_SDK_MGCL_API_Reference	API for the MGCL.

Typographic Conventions

Certain Typographic Conventions are used throughout this manual to clarify the meaning of the text:

Convention	Meaning
<code>courier</code>	Indicates literal program code.
<i>italic</i>	Indicates names of parameters and structure members (in structure/function definitions only).
bold	Indicates data types and structure/function names (in structure/function definitions only).
<code>blue</code>	Indicates function name.
blue	Indicates a hyperlink.

Developer Support

Sony Computer Entertainment America (SCEA)

SCEA developer support is available to licensees in North America only. You may obtain developer support or additional copies of this documentation by contacting the following addresses:

Order Information	Developer Support
Attn: Developer Tools Coordinator Sony Computer Entertainment America 919 East Hillsdale Blvd. Foster City, CA 94404, U.S.A. Tel: (650) 655-8000	E-mail: scert-support@scea.com scea_support@ps2-pro.com Web: https://www.ps2-pro.com/ https://psp.scedev.net Developer Support Hotline: (650) 655-5566 (Call Monday through Friday, 8 a.m. to 5 p.m., PST/PDT)

Sony Computer Entertainment Europe (SCEE)

SCEE developer support is available to licensees only in the PAL television territories (including Europe and Australasia). You may obtain developer support or additional copies of this documentation by contacting the following addresses:

Order Information	Developer Support
Attn: Production Coordinator Sony Computer Entertainment Europe 13 Great Marlborough Street London W1F 7HP, U.K. Tel: +44 (0) 20 7859-5000	E-mail: scee_support@ps2-pro.com Web: https://www.ps2-pro.com/ https://psp.scedev.net Developer Support Hotline: +44 (0) 20 7911-7711 (Call Monday through Friday, 9 a.m. to 6 p.m., GMT/BST)

Chapter 1: Defines/Macros

This page intentionally left blank.

MAX_CLIENTS_PER_WORLD

This is the maximum number of clients per game world.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	October 1, 2001

Syntax

```
#define MAX_CLIENTS_PER_WORLD 256
```

Description

This is the maximum number of clients per game world supported by this MGCL client.

Notes

N/A

Example

N/A

See also

N/A

MEDIUS_GAME_COMM_LIBRARY_VERSION_NUMBER_MAXLEN

Maximum number of bytes used to store the MGCL version string literal, including NULL termination.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	October 1, 2001

Syntax

```
#define MEDIUS_GAME_COMM_LIBRARY_VERSION_NUMBER_MAXLEN 64
```

Description

Maximum number of bytes used to store the MGCL version string literal, including NULL termination.

Notes

N/A

Example

N/A

See also

[MGCLGetMGCLVersion\(\)](#)

MGCL_ACCESSKEY_MAXLEN

Maximum number of bytes used to represent the server's access key, which is used when the server authorizes a new client to connect.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	October 1, 2001

Syntax

```
#define MGCL_ACCESSKEY_MAXLEN 17
```

Description

Maximum number of bytes used to represent the game server's access key.

Notes

N/A

Example

N/A

See also

N/A

MGCL_GAMENAME_MAXLEN

Maximum number of bytes that a game name may contain, including the NULL termination.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	October 1, 2001

Syntax

```
#define MGCL_GAMENAME_MAXLEN 64
```

Description

Maximum number of bytes used for a game name, including the NULL termination.

Notes

Game names should use the same character set encoding as the Medius Client login.

Example

N/A

See also

N/A

MGCL_GAMEPASSWORD_MAXLEN

Maximum number of bytes that a game password can contain, including the NULL termination.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	October 1, 2001

Syntax

```
#define MGCL_GAMEPASSWORD_MAXLEN 32
```

Description

Maximum number of bytes that a game password can contain, including the NULL termination.

Notes

N/A

Example

N/A

See also

N/A

MGCL_GAMESTATS_MAXLEN

Maximum number of bytes in the game-stats string, including the NULL termination.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	October 1, 2001

Syntax

```
#define MGCL_GAMESTATS_MAXLEN 256
```

Description

Maximum number of bytes in the game-stats string, including the NULL termination.

Notes

N/A

Example

N/A

See also

N/A

MGCL_MESSAGEID_MAXLEN

Maximum number of bytes for the message ID field, including the NULL termination.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	October 1, 2001

Syntax

```
#define MGCL_MESSAGEID_MAXLEN 21
```

Description

Maximum number of bytes for the message ID field, including the NULL termination. This value is used for all MessageID fields in request structures, and can be used to match up an asynchronous request with the response value.

Notes

N/A

Example

N/A

See also

N/A

MGCL_SERVERIP_MAXLEN

Maximum number of bytes in the server's IP Address string, including the NULL termination.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	October 1, 2001

Syntax

```
#define MGCL_SERVERIP_MAXLEN 20
```

Description

The maximum number of bytes used to represent the game servers IP address.

Notes

N/A

Example

N/A

See also

N/A

MGCL_SERVERPORT_MAXLEN

Maximum number of bytes in the server's port number string, including the NULL termination.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	October 1, 2001

Syntax

```
#define MGCL_SERVERPORT_MAXLEN 8
```

Description

Maximum number of bytes in the server's port number string, including the NULL termination.

Notes

Not used in version 2.10.

Example

N/A

See also

N/A

MGCL_SERVERVERSION_MAXLEN

Maximum number of bytes in the version string literal of MGCL.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	October 1, 2001

Syntax

```
#define MGCL_SERVERVERSION_MAXLEN 16
```

Description

Maximum number of bytes in the version string literal of MGCL, including the NULL termination.

Notes

N/A

Example

N/A

See also

N/A

MGCL_SESSIONKEY_MAXLEN

Maximum number of bytes for the session key field, including the NULL termination.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	October 1, 2001

Syntax

```
#define MGCL_SESSIONKEY_MAXLEN 17
```

Description

Maximum number of bytes in the session key field, including the NULL termination.

Notes

N/A

Example

N/A

See also

N/A

MGCL_VERSION_BUILD

Current MGCL build number.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	October 1, 2001

Syntax

```
#define MGCL_VERSION_BUILD 2
```

Description

Current MGCL version build number.

Notes

This is used in with the major and minor version numbers.

Example

N/A

See also

N/A

MGCL_VERSION_MAJOR

Current MGCL major version number.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	October 1, 2001

Syntax

```
#define MGCL_VERSION_MAJOR 2
```

Description

Current MGCL major version number.

Notes

This is used in conjunction with the minor and build version numbers.

Example

N/A

See also

N/A

MGCL_VERSION_MINOR

Current MGCL minor version number.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	October 1, 2001

Syntax

```
#define MGCL_VERSION_MINOR 10
```

Description

Current MGCL minor version number.

Notes

This is used in conjunction with the major and build version numbers.

Example

N/A

See also

N/A

Chapter 2: Enumerated Types

This page intentionally left blank.

DMEInitStatus

Indicates whether or not DME has already been initialized.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	October 31, 2001

Enumeration

```
typedef enum {
    DMEInitialized = 0,
    DMENotInitialized = 1,
    ExtraDMEInitStatus = 0xfffff
} DMEInitStatus;
```

The DME has been initialized prior to this function call.

The DME has NOT been initialized prior to this function call.

Vendor-specific compiler placeholder. This is never used explicitly.

Description

Enumeration used to identify whether or not the DME layer has been initialized prior to the call to MGCLInitialize.

Notes

There is no default value in MGCLInitialize for this parameter.

Example

```
MGCLInitializeInParams stMGCLInit;
MGCLInitializeOutParams stMGCLInitResult;
stMGCLInit.Status = DMENotInitialized;
... set other parameters ...
MGCLInitialize(&stMGCLInit, &stMGCLInitResult);
```

See also

[MGCLInitializeInParams](#)

MGCL_ALERT_LEVEL

This is an enumeration used to identify the current level of service. This prevents create/join games from occurring on this host.

Link to file	Include file	Introduced	Last modified
librtmgcl.a	mgcl.h	1.0	October 31, 2001

Enumeration

```
typedef enum {
    MGCL_ALERT_NONE,                Default, no alert occurs. Allow normal use.
    MGCL_ALERT_LOW,                 Low load. No new clients may join.
    MGCL_ALERT_MID,                 Moderate load. No new clients may join.
    MGCL_ALERT_HIGH,                High load. No new clients may join.
    ExtraMGCLAlertCode = 0xfffff,   Vendor-specific compiler placeholder. This is never
                                     used explicitly.
} MGCL_ALERT_LEVEL;
```

Description

This is an enumeration used to indicate the current level of service. This allows the MGCL host to "raise an alert", at which point no more create/join games are permitted on that MGCL instance until the alert level goes back to an acceptable level.

Notes

The alert is MGCL_ALERT_NONE for normal behavior, or it is MGCL_ALERT_LOW, MGCL_ALERT_MID or MGCL_ALERT_HIGH for abnormal behavior.

Example

```
MediusServerReportType myServerReport;
myServerReport.AlertLevel = MGCL_ALERT_NONE;
MGCLSendServerReport (&myServerReport);
```

See also

[MediusServerReportType](#)

MGCL_ERROR_CODE

Enumeration used to identify the response status within a callback, as well as from a function call.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	October 31, 2001

Enumeration

typedef enum {	
MGCL_SUCCESS = 0,	SUCCESSFUL response.
MGCL_CONNECTION_ERROR = -1,	Connect terminated.
MGCL_CONNECTION_FAILED = -2,	Unable to connect to a target host.
MGCL_DISCONNECT_FAILED = -3,	Unable to disconnect from a target host.
MGCL_NOT_CONNECTED = -4,	Attempt to use an API call that requires a connection - without a connection.
MGCL_SEND_FAILED = -5,	Sending of data failed.
MGCL_INITIALIZATION_FAILED = -6,	Initialization of the MGCL library failed.
MGCL_SHUTDOWN_ERROR = -7,	Shutdown of the MGCL library failed.
MGCL_NETWORK_ERROR = -8,	A lower level network error occurred.
MGCL_AUTHENTICATION_FAILED = -9,	Authentication of the MGCL host failed. This may be due to application ID or mismatched security keys.
MGCL_SESSIONBEGIN_FAILED = -10,	Session begin failed.
MGCL_SESSIONEND_FAILED = -11,	Session end failed.
MGCL_UNSUCCESSFUL = -12,	General request failed.
MGCL_INVALID_ARG = -13,	An invalid argument was used in a function call.
MGCL_NATRESOLVE_FAILED = -14,	Unable to access the NAT service or resolve the internal NAT address.
MGCL_GAME_NAME_EXISTS = -15,	A game with the same name already exists.
MGCL_WORLDID_INUSE = -16,	The specified world ID is already in use.
MGCL_DME_ERROR = -17,	A lower level DME error has occurred.
MGCL_CALL_MGCL_CLOSE_BEFORE_REINITIALIZING = -18,	An attempt was made to re-initialize MGCL without first closing the subsystem.
MGCL_NUM_GAME_WORLDS_PER_LOBBY_WORLD_EXCEEDED = -19,	The maximum number of games within a lobby world was exceeded.
ExtraMGCLStatusCode = 0xfffff	Vendor-specific compiler placeholder. This is never explicitly used.
} MGCL_ERROR_CODE;	

Description

This enumeration identifies the response status within a callback. All callback status codes should be one of these enumerations. MGCL_ERROR_CODE can be returned by any MGCL function call.

Notes

This enum is used in every callback response, as well as the return code from function calls.

Example

```
MGCL_ERROR_CODE returnStatus;  
returnStatus = MGCLClose();
```

See also

N/A

MGCL_EVENT_TYPE

This enumeration specifies the type of connect event that is sent to Medius in an event notification message.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	October 31, 2001

Enumeration

```
typedef enum {
    MGCL_EVENT_CLIENT_DISCONNECT = 0,           A client disconnected from this game server.
    MGCL_EVENT_CLIENT_CONNECT = 1,             A server connected to this game server.
    MGCL_EVENT_EXTRA = 0xFFFFF                Vendor-specific compiler placeholder. This is never
                                              used explicitly.
} MGCL_EVENT_TYPE;
```

Description

This enumeration specifies the type of connect event that is sent to Medius in an event notification message.

Notes

This is used when a client connects or disconnects from this host. This has ramifications in the player count as known by the server.

Example

```
MediusServerConnectNotificationType stConnectNotification;
stConnectNotification.ConnectEventType = MGCL_EVENT_CLIENT_CONNECT;
```

See also

[MediusServerConnectNotificationType](#), [MGCLSendConnectNotification](#)

MGCL_GAME_HOST_TYPE

Specify which type of game host this MGCL application is running.

Link to file	Include file	Introduced	Last modified
librtmgcl.a	mgcl.h	1.0	October 31, 2001

Enumeration

```
typedef enum {
  MGCLGameHostClientServer = 0,
  MGCLGameHostIntegratedServer = 1,
  MGCLGameHostPeerToPeer = 2,
  MGCLGameHostLANPlay = 3,
  MGCLGameHostClientServerAuxUDP = 4,
  ExtraMGCLGameHost = 0xfffff
} MGCL_GAME_HOST_TYPE;
```

The game server is configured for client-server gaming.

The game server is configured for an integrated server with both game play and serving.

The game server is configured for the host in a peer-to-peer game.

This is the host of a LAN game.

This game server is configured for a client-server auxilliary UDP gaming.

Vendor-specific compiler placeholder. This is never used explicitly.

Description

This is the enumeration used to identify the game type to be created or the host type.

Notes

A host can only specify a single type of game. ClientServer is different from ClientServerAuxUDP

Example

```
MediusServerCreateGameOnMeRequest stCreateGameRequest;
stCreateGameRequest.GameHostType = MGCLGameHostPeerToPeer;
```

See also

[MediusServerCreateGameOnMeRequest](#), [MediusServerSessionBeginRequest](#)

MGCL_SERVER_ATTRIBUTES

This enumeration determines the specific attributes of this server during server authentication.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	October 31, 2001

Enumeration

```
typedef enum {
    MGCL_SERVER_NONE = 0,
    MGCL_SERVER_ALLOW_REBROADCAST = (1 << 0),

    MGCL_SERVER_ALLOW_SPECTATOR = (1 << 1),
    MGCL_SERVER_ALLOW_INFORMER = (1 << 2),

    MGCL_SERVER_ALLOW_MONITOR = (1 << 3),
    MGCL_SERVER_EXTRA = 0xFFFFF
} MGCL_SERVER_ATTRIBUTES;
```

This server has no special attributes.

This server allows for the rebroadcasting of game data.

This server supports spectators to receive data.

This server can be used as an informer type of server. The description is ambiguous on purpose.

This server can be used to monitor game traffic.

Vendor-specific compiler placeholder. This is never used explicitly.

Description

This enumeration determines the specific attributes of this server during server authentication

Notes

This enumeration is used as a bit-field with values logically OR'ed together.

Example

```
MediusServerSetAttributesRequest stMyServerAttributes = 0;
stMyServerAttributes.Attributes |= MGCL_SERVER_ALLOW_SPECTATOR;
```

See also

[MGCLSetServerAttributes](#), [MediusServerSetAttributesRequest](#)

MGCL_TRUST_LEVEL

This enumeration identifies the trust level for this host.

Link to file	Include file	Introduced	Last modified
librtmgcl.a	mgcl.h	1.0	October 31, 2001

Enumeration

```
typedef enum {
    MGCL_TRUSTED = 0,
    MGCL_NOT_TRUSTED = 1,

    ExtraMGCLTrustLevelCode = 0xfffff
} MGCL_TRUST_LEVEL;
```

This server is a trusted game server.

This server is NOT a trusted game server. This is used for all peer-to-peer game hosts.

Vendor-specific compiler placeholder. This is never used explicitly.

Description

Indicate whether you are an internal server or a public untrusted user.

- Trusted = DME Servers
- NotTrusted = PeerToPeer
- NotTrusted = Integrated Servers

Notes

Deprecated. Only internal servers should use the MGCL_TRUSTED setting.

Example

```
MediusServerAuthenticationRequest stAuthRequest;
stAuthRequest.TrustLevel = MGCL_NOT_TRUSTED;
```

See also

[MediusServerAuthenticationRequest](#)

MGCL_WORLD_ATTRIBUTES

This enumeration identifies the attributes of the game world to be created.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	2.9	October 31, 2001

Enumeration

```
typedef enum {
    MGCL_WORLD_NONE = 0,
    MGCL_WORLD_TYPE_SPECTATOR = (1 << 0),
    MGCL_WORLD_ALLOW_SPECTATORS = (1 << 1),
    MGCL_WORLD_INTERNAL = (1 << 2),
    MGCL_WORLD_TYPE_INFORMER = (1 << 3),
    MGCL_WORLD_EXTRA = 0xFFFFF
} MGCL_WORLD_ATTRIBUTES;
```

This game world has no specific attributes.

This game world is a spectator game world, and has no input from any client.

This game world can support spectators that want to come and watch.

This is an internal game world; it is NOT for general use.

This game world is an informer game world type. Sheriffs may visit it.

Vendor-specific compiler placeholder. This is never used explicitly.

Description

This enumeration identifies the attributes of the game world to be created.

Notes

This enum is actually a bitfield. It is an alternative callback request to create a game world. Normal game worlds arrive from the following callback: `MGCLServerCreateGameCallback` `MyServerCreateGameCallback`; located inside [MGCLInitializeInParams](#).

Example

N/A

See also

[MediusServerCreateGameWithAttributesRequest](#), [MGCLServerCreateGameWithAttributesCallback](#), [MGCLInitializeInParams](#)

MGCL_WORLD_STATUS

This enumeration indicates the appropriate world status in world reports.

Link to file	Include file	Introduced	Last modified
librtmgcl.a	mgcl.h	1.0	October 31, 2001

Enumeration

```
typedef enum {
    MGCL_WORLD_INACTIVE = 0,
    MGCL_WORLD_STAGING = 1,

    MGCL_WORLD_ACTIVE = 2,
    MGCL_WORLD_CLOSED = 3,

    ExtraMGCLWorldStatusCode = 0xffff

} MGCL_WORLD_STATUS;
```

This game world is NOT active.

This game world has people staging, but not yet in the game.

This game world is already in progress.

This game world is no longer taking new clients. It is closed.

Vendor-specific compiler placeholder. This is never used explicitly.

Description

This enumeration indicates the appropriate world status in world reports.

Notes

This is helpful in game filtering to denote when people are getting ready (staging), or are already playing a game (active).

Example

```
MediusServerWorldReportOnMe myWorldReport;
myWorldReport.WorldStatus = MGCL_WORLD_STAGING;
```

See also

[MediusServerWorldReportOnMe](#)

Chapter 3: Structures

This page intentionally left blank.

MediusServerAuthenticationRequest

Request "authentication" of this server via [MGCLAuthenticationSend\(\)](#).

Link to file	Include file	Introduced	Last modified
librtmgcl.a	mgcl.h	1.0	October 31, 2001

Structure

```
typedef struct {
    char MessageID[MGCL_MESSAGEID_MAXLEN];           Message ID used for asynchronous request
                                                       processing.
    MGCL\_TRUST\_LEVEL TrustLevel;                     Trust level for this game server.
    NetAddressList AddressList;                       Server address or port for standalone GS; it is only
                                                       populated internally by MGCL.
} MediusServerAuthenticationRequest;
```

Description

Request "authentication" of this server via [MGCLAuthenticationSend\(\)](#).

Notes

This structure should always populate the TrustLevel field as MGCL_NOT_TRUSTED for all peer-to-peer titles.

Example

```
MGCL_ERROR_CODE mgcl_result;
MediusServerAuthenticationRequest stServerAuth;
memset(&stServerAuth, 0, sizeof(stServerAuth));
sprintf(stServerAuth.MessageID, "1");
stServerAuth.TrustLevel = MGCL_NOT_TRUSTED;
mgcl_result = MGCLAuthenticationSend(&stServerAuth,
                                     MyMGCLServerAuthenticationCallback,
                                     NULL);
```

See also

[MGCLAuthenticationSend\(\)](#)

MediusServerAuthenticationResponse

Data in the response callback for [MGCLAuthenticationSend\(\)](#).

Link to file	Include file	Introduced	Last modified
librtmgcl.a	mgcl.h	1.0	October 31, 2001

Structure

```
typedef struct {
    char MessageID[MGCL_MESSAGEID_MAXLEN];           Message ID used for asynchronous request
                                                       processing.
    char Confirmation;                                MGCL_SUCCESS or an error.
    NetConnectionInfo ConnectInfo;                    Address of the proxy server to connect to.
} MediusServerAuthenticationResponse;
```

Description

Response structure used in the response callback for [MGCLAuthenticationSend\(\)](#).

Notes

The Confirmation value maps to MGCL_ERROR_CODE. If it is MGCL_SUCCESS, then the ConnectInfo points to the proxy server to which this host can connect in order to interface with the Medius universe. Save the connect information in a variable, and use it with MGCL_Connect().

Example

```
MediusServerAuthenticationResponse *pstAuthResponse;
int iTimeout=600;
MGCLAuthenticationSend(&stServerAuth, MyMGCLServerAuthenticationCallback,
                      NULL);
while(bAuthCallback || !iTimeout) {MGCLUpdate(); iTimeout--;}
```

See also

[MGCLAuthenticationSend\(\)](#)

MediusServerConnectGamesRequest

Request from the Medius Servers to connect a game world to this host.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	February 1, 2002

Structure

```
typedef struct {
    char MessageID[MGCL_MESSAGEID_MAXLEN];           Message ID used for asynchronous request
                                                       processing.
    char ServerIP[MGCL_SERVERIP_MAXLEN];              IP address of the server to connect to.
    int ServerPort;                                    Port for the server to connect to.
    int GameWorldID;                                   Game world ID to connect to.
    int SpectatorWorldID;                             Specator world ID to connect to.
} MediusServerConnectGamesRequest;
```

Description

Request from the Medius Servers to connect a game world to this host.

Notes

This is a request sent by the unsolicited callback defined in MGCLInitializeInParams. The request is for spectator worlds to be able to connect to this host. All peer-to-peer titles should not need to reference this structure. Only client-server spectator game servers need to use this to connect themselves to another game broadcast server.

Example

None.

See also

[MGCLServerConnectGamesCallback\(\)](#), [MGCLInitializeInParams](#)

MediusServerConnectGamesResponse

Response to Medius servers with the status of a particular world and whether it is allowed to be connected to this host.

Link to file	Include file	Introduced	Last modified
librtmgcl.a	mgcl.h	2.9	November 1, 2004

Structure

```

typedef struct {
    char MessageID[MGCL_MESSAGEID_MAXLEN];           Message ID used for asynchronous request
                                                       processing.
    int GameWorldID;                                  Game world ID of the world being requested.
    int SpectatorWorldID;                             Spectator world ID of the world being requested.
    char Confirmation;                                MGCL_SUCCESS or other code to indicate an
                                                       error to Medius.
} MediusServerConnectGamesResponse;

```

Description

This is a response structure used in [MGCLConnectGamesResponse\(\)](#). This is related to the connection of spectator game worlds to this host, and is not a general peer-to-peer function.

MediusServer -> UnsolicitedCB (MyServerConnectGamesCallback defined at MGCL initialization time) -> Save Data. Use this to respond to the request.

Notes

The confirmation maps to MGCL_ERROR_CODE. It is set to MGCL_SUCCESS to allow the world creation to occur. It is usually used for client/server games, and not for peer-to-peer games.

Example

```

MediusServerConnectGamesResponse myServerConnectGamesResponse;
myServerConnectGamesResponse.GameWorldID = InboundGameWorldID;
myServerConnectGamesResponse.SpectatorWorldID = InboundSpectatorWorldID;
myServerConnectGamesResponse.SpectatorWorldID = MGCL_SUCCESS;
MGCLConnectGamesResponse (&myServerConnectGamesResponse) ;

```

See also

[MGCLServerConnectGamesCallback](#), [MGCLConnectGamesResponse\(\)](#)

MediusServerConnectNotificationType

Request a structure to notify Medius about the connect or disconnect of a client on this game host.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	2.9	November 1, 2004

Structure

```
typedef struct {
    MGCL\_EVENT\_TYPE ConnectEventType;           A connect or disconnect event.
    unsigned int MediusWorldUID;                 Medius game world unique ID that the player
                                                connected or disconnected from.

    char PlayerSessionKey[MGCL_SESSIONKEY_MAXLEN]; The player's session key.
} MediusServerConnectNotificationType;
```

Description

Structure used when calling [MGCLSendConnectNotification\(\)](#)

Notes

This is used as a fire-and-forget message.

Example

```
MediusServerConnectNotificationType myServerConnectNotification;
myServerConnectNotification.ConnectEventType = MGCL_EVENT_CLIENT_CONNECT;
myServerConnectNotification.MediusWorldUID = WorldIDOfNewlyConnectedClient.
memcpy(myServerConnectNotification.PlayerSessionKey, acClientsSessionKey,
        MGCL_SESSIONKEY_MAXLEN);
errorCode = MGCLSendConnectNotification(&myServerConnectNotification);
```

See also

[MGCLSendConnectNotification\(\)](#)

MediusServerCreateGameOnMeRequest

Request a structure to create a game on this game server using [MGCLCreateGameOnMeRequest\(\)](#).

Link to file	Include file	Introduced	Last modified
librtmgcl.a	mgcl.h	1.0	October 31, 2001

Structure

typedef struct {	
char <i>MessageID</i> [MGCL_MESSAGEID_MAXLEN];	Message ID used for asynchronous request processing.
char <i>GameName</i> [MGCL_GAMENAME_MAXLEN];	Name of the game to create host.
char <i>GameStats</i> [MGCL_GAMESTATS_MAXLEN];	Initial game statistics, an ASCII NULL terminated string.
char <i>GamePassword</i> [MGCL_GAMEPASSWORD_MAXLEN];	Game password, including NULL termination.
int <i>ApplicationID</i> ;	Application ID of this game
int <i>MaxClients</i> ;	Maximum number of clients.
int <i>MinClients</i> ;	Minimum number of clients to advertise to others when they get the game lists.
int <i>GameLevel</i> ;	Game level, title dependent.
int <i>PlayerSkillLevel</i> ;	General player skill level, title dependent.
int <i>RulesSet</i> ;	Rule set, title dependent.
int <i>GenericField1</i> ;	Generic field, used for filtering, title dependent.
int <i>GenericField2</i> ;	Generic field, used for filtering, title dependent.
int <i>GenericField3</i> ;	Generic field, used for filtering, title dependent.
int <i>GenericField4</i> ;	Generic field, used for filtering, title dependent.
int <i>GenericField5</i> ;	Generic field, used for filtering, title dependent.
int <i>GenericField6</i> ;	Generic field, used for filtering, title dependent.
int <i>GenericField7</i> ;	Generic field, used for filtering, title dependent.
int <i>GenericField8</i> ;	Generic field, used for filtering, title dependent.
MGCL_GAME_HOST_TYPE <i>GameHostType</i> ;	Populated internally by MGCL based on the <i>ServerType</i> set in <i>ServerSessionBegin</i>
NetAddressList <i>AddressList</i> ;	Address list for the peer-to-peer host; populated internally by MGCL.
int <i>WorldID</i> ;	Passed in. It is a standalone that can determine which world ID to use, and used for peer-to-peer integrated
int <i>AccountID</i> ;	The creator's account ID. This ID is used to tie the game to the current user's lobby world.
} MediusServerCreateGameOnMeRequest;	

Description

Request a structure to create a game on this game server using [MGCLCreateGameOnMeRequest\(\)](#). Ensure that this host has already authenticated itself as a game server, and that it is sending server reports (as a heartbeat) on a periodic basis.

Notes

The Confirmation value maps to MGCL_ERROR_CODE.

Example

```
. Call NetHostPeerToPeer(), and save the output parameter.
.
MGCLCreateGameOnMeRequestInParams myCreateGameOnMeRequestInParams;
myCreateGameOnMeRequestInParams CreateGameRequest.MinClients = 2;
. Fill in the other fields.
.
myCreateGameOnMeRequestInParams.CreateGameCallback = myCreateGameOnMeCallback;
.
. Use the output parameter from NetHostPeerToPeer() here.
myCreateGameOnMeRequestInParams.ConnectionHandle =
    myNetHostPeerToPeerOutParams.ConnectionHandle;
```

See also

[MGCLCreateGameOnMeRequest\(\)](#), [MGCLCreateGameOnMeRequestInParams](#)

MediusServerCreateGameOnMeResponse

Response structure for the request to create a game using a callback invoked by [MGCLCreateGameOnMeRequest\(\)](#).

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.00	October 31, 2001

Structure

```
typedef struct {
    char MessageID[MGCL_MESSAGEID_MAXLEN];    Message ID used for asynchronous request
                                                processing.
    char Confirmation;                        MGCL_SUCCESS or a MGCL_ERROR_CODE
                                                error code.
    int MediusWorldID;                       Game world ID assigned by Medius.
} MediusServerCreateGameOnMeResponse;
```

Description

Response structure for the request to create a game using a callback invoked by [MGCLCreateGameOnMeRequest\(\)](#)

Notes

Calling MediusGetGames() will return a list of games, each uniquely identified by a MediusWorldID. The peer-to-peer host uses the MediusWorldID for WorldReports and PlayerReports. To join peer-to-peer Clients, use the MediusWorldID to perform a MediusJoinGame() with and for their Player reports.

Example

```
void MyMGCLCreateGameOnMeRequestCallback(
    MediusServerCreateGameOnMeResponse* pResponse,
    void* arg)
{
    if(pResponse->Confirmation == MGCL_SUCCESS)
    {
        // Store the Medius game world ID for use in world reports, or
        // other game related API functions.
        G_MyMediusGameWorldID = pResponse->MediusWorldID;
    }
    else
    {
        // This could be due to the fact that the game server (in the case
        // of a peer-to-peer host that is the PS itself) has timed out due
        // to a failure to periodically send server reports via MGCL.
        // Then, the game server is removed from Medius, and all attempts
        // to create games will fail. Then, a re-authentication is
        // necessary.
        printf("Error creating game on self.\n");
    }
}
```

See also

[MGCLCreateGameOnMeRequest\(\)](#)

MediusServerCreateGameRequest

MediusServerCreateGameRequest is used within the MGCLServerCreateGameCallback.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	February 1, 2002

Structure

```
typedef struct {
    char MessageID[MGCL_MESSAGEID_MAXLEN];
    int ApplicationID;
    int MaxClients;
} MediusServerCreateGameRequest;
```

Message ID used for asynchronous request processing.

Application ID of the client that wants to create a game on this host.

Maximum number of clients that the game world is expected to contain.

Description

Informs this game server of an incoming request by a user to create a game on this game server host

Notes

This structure is only used in the context of the callback. Respond with [MGCLCreateGameResponse\(\)](#).

Example

In the context of:

```
myMGCLServerCreateGameCallback (MediusServerCreateGameRequest *ThisPacket,
                                void* pUserData)
```

Use the information about the application ID and maximum number of clients to determine if this game host has enough resources to support the application and number of clients.

See also

[MGCLServerCreateGameCallback\(\)](#), [MGCLInitializeInParams](#)

MediusServerCreateGameResponse

Response that a game world was created on this game server.

Link to file	Include file	Introduced	Last modified
librtmgcl.a	mgcl.h	1.0	February 1, 2002

Structure

```

typedef struct {
    char MessageID[MGCL_MESSAGEID_MAXLEN];           Message ID used for asynchronous request
                                                       processing.
    char Confirmation;                                MGCL_SUCCESS or other value to indicate an
                                                       error.
    int WorldID;                                       World ID of the game hosted on this server.
} MediusServerCreateGameResponse;

```

Description

Response that a game world was created on this game server. MediusServerCreateGameResponse is used by [MGCLCreateGameResponse\(\)](#)

Notes

Response to grant or deny game creation on this MGCL host. The confirmation maps to MGCL_ERROR_CODE.

Example

```

MediusServerCreateGameResponse myServerCreateGameResponse;
myServerCreateGameResponse.Confirmation = MGCL_SUCCESS;

```

See also

[MGCLCreateGameResponse\(\)](#)

MediusServerCreateGameWithAttributesRequest

MediusServerCreateGameWithAttributesRequest is used within the MGCLServerCreateGameWithAttributesCallback.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	2.9	November 1, 2004

Structure

```
typedef struct {
    char MessageID[MGCL_MESSAGEID_MAXLEN];
    int ApplicationID;
    int MaxClients;
    int Attributes;
    unsigned int MediusWorldUID;
} MediusServerCreateGameWithAttributesRequest;
```

Message ID used for asynchronous request processing.

Application ID of the client that wants to create a game on this host.

Maximum number of clients the game world is can contain.

MGCL specific attributes.

Medius game world unique ID.

Description

Informs this game server of an incoming request by a user to create a game on this game server host with a specific set of attributes.

Notes

This structure is only used in a callback context. Respond with [MGCLCreateGameWithAttributesResponse\(\)](#)

Example

In the context of

```
myMGCLServerCreateGameWithAttributesCallback
    (MediusServerCreateGameWithAttributesRequest *ThisPacket,
     void* pUserData)
```

Use the information about the application ID and maximum number of clients to determine if this game host has enough resources to support the application and number of clients.

See also

[MGCLServerCreateGameWithAttributesCallback\(\)](#), [MGCLInitializeInParams](#)

MediusServerCreateGameWithAttributesResponse

MediusServerCreateGameWithAttributesResponse is used by [MGCLCreateGameWithAttributesResponse\(\)](#).

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	2.9	November 1, 2004

Structure

```
typedef struct {
    char MessageID[MGCL_MESSAGEID_MAXLEN];    Message ID used for asynchronous request
                                                processing.
    char Confirmation;                         MGCL_SUCCESS or other value to indicate an
                                                error.
    int WorldID;                               Game world ID of the game created on this host.
} MediusServerCreateGameWithAttributesResponse;
```

Description

Response that a game world was created on this game server.

Notes

Response to grant or deny game creation on this MGCL host. The confirmation maps to MGCL_ERROR_CODE. This supercedes the previous function [MGCLCreateGameResponse\(\)](#).

Example

```
MediusServerCreateGameWithAttributesResponse myServerCreateGameResponse;
unsigned short world_index=0;

myServerCreateGameWithAttributesResponse.Confirmation = MGCL_SUCCESS;
DmeServerGetNextWorld(&world_index);
DmeServerInitializeWorld(world_index, MaxClients, ...);

myServerCreateGameWithAttributesResponse.WorldID = world_index;

MGCLCreateGameWithAttributesResponse(&myServerCreateGameWithAttributesResponse);
```

See also

[MGCLCreateGameWithAttributesResponse\(\)](#)

MediusServerDisconnectPlayerRequest

Request a structure to forcefully disconnect a player from a game world on this host.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	2.10	February 1, 2005

Structure

```
typedef struct {
    int DmeWorldID;           DME world ID of the player connected to this host.
    int DmeClientIndex;       DME client index of the player connected to this
                              host.
} MediusServerDisconnectPlayerRequest;
```

Description

Request a structure used when the Medius server requests a player to be disconnected from this host by the unsolicited callback. MediusServer -> UnsolicitedCB (MyServerDisconnectPlayerCallback defined at MGCL initialization time) -> MGCLServerDisconnectPlayerCallback

Notes

The host is responsible for disconnecting the client from the world ID and client index using a DME server library call.

Example

```
MGCLServerDisconnectPlayerCallback()
(MediusServerDisconnectPlayerRequest *ThisPacket, void *pUserData)
DmeServerSendForcedClientDisconnectMsg(
    ThisPacket->DmeWorldID, ThisPacket->DmeClientIndex,
    DME_SERVER_FORCED_DISCONNECT_BANNED);
```

See also

[MGCLInitializeInParams](#), [MGCLInitialize](#)

MediusServerEndGameOnMeRequest

Request by this MGCL host server to end a game on this host.

Link to file	Include file	Introduced	Last modified
librtmgcl.a	mgcl.h	1.0	February 1, 2002

Structure

```
typedef struct {
    char MessageID[MGCL_MESSAGEID_MAXLEN];           Message ID used for asynchronous request
                                                       processing.
    int MediusWorldID;                                World ID of the game world to end.
} MediusServerEndGameOnMeRequest;
```

Description

Request structure used when calling [MGCLEndGameOnMeRequest\(\)](#)

Notes

This structure uses the game world ID as MediusWorldID. This should not be confused with the net world ID on this host.

Example

```
MediusServerEndGameOnMeRequest myServerEndGameOnMeRequest;
myServerEndGameOnMeRequest.MediusWorldID = G_MyMediusGameWorldID;
MGCLEndGameOnMeRequest (&myServerEndGameOnMeRequest, NULL);
```

See also

[MGCLEndGameOnMeRequest](#)

MediusServerEndGameOnMeResponse

Response status denoting whether or not the "MediusServerEndGameOnMeRequest" function was successful.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	February 1, 2002

Structure

```
typedef struct {
    char MessageID[MGCL_MESSAGEID_MAXLEN];    Message ID used for asynchronous request
                                                processing.
    char Confirmation;                         MGCL_SUCCESS or other value to indicate an
                                                error.
} MediusServerEndGameOnMeResponse;
```

Description

Response structure used in the callback from [MGCLEndGameOnMeRequest\(\)](#).

Notes

The confirmation maps to MGCL_ERROR_CODE.

Example

```
(MGCLServerEndGameOnMeCallback)
(MediusServerEndGameOnMeResponse *ThisPacket, void* pUserData)
assert(ThisPacket->Confirmation == MGCL_SUCCESS);
```

See also

[MGCLEndGameOnMeRequest](#)

MediusServerEndGameRequest

A Medius server is telling this host to end a game.

Link to file	Include file	Introduced	Last modified
librtmgcl.a	mgcl.h	1.0	February 1, 2002

Structure

```

typedef struct {
    char MessageID[MGCL_MESSAGEID_MAXLEN];           Message ID used for asynchronous request
                                                       processing.
    int WorldID;                                       World ID of the game to kill.
    char BrutalFlag;                                   Boolean, to either Kill now, or allow the game to
                                                       finish and then destroy the game world.
} MediusServerEndGameRequest;

```

Description

A Medius server is telling this host to end a game. It is invoked through the callback defined at MGCL initialization time. Peer-to-peer clients usually do not need to consider this request.

Notes

The BrutalFlag is either 0 or 1. Under normal circumstances, the game should end when the client count for the game world reaches zero. If the brutal flag is True, then the world should be destroyed immediately, and all of the clients forcefully disconnected.

Example

In the context of

```

myMGCLServerEndGameCallback
    (MediusServerEndGameRequest *ThisPacket, void* pUserData)

if (ThisPacket->BrutalFlag) {
    DME_SERVER_RESULT rc = DmeServerSendForcedWorldDisconnectMsg(WorldID,
                                                                DME_SERVER_FORCED_DISCONNECT_END_SESSION);
    rc = DmeServerCleanupWorld(WorldID);
} else {
    unsigned short client_count;
    DME_SERVER_RESULT rc = DmeServerGetNumClients(WorldID, &client_count);
    if (client_count == 0) {
        rc = DmeServerCleanupWorld(WorldID);
    } else {
        DmeServerSetWorldSelfDestruct(WorldID);
    }
}

```

See also

[MGCLServerEndGameCallback](#), [MGCLInitializeInParams](#)

MediusServerEndGameResponse

Response from this host to denote that a game world has been ended.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	February 1, 2002

Structure

```
typedef struct {
    char MessageID[MGCL_MESSAGEID_MAXLEN];    Message ID used for asynchronous request
                                                processing.
    char Confirmation;                          MGCL_SUCCESS or other value to indicate an
                                                error.
} MediusServerEndGameResponse;
```

Description

Response for an end game request. This structure is used in MGCLEndGameResponse

Notes

The confirmation maps to MGCL_ERROR_CODE. If it is MGCL_SUCCESS, then cleanup has occurred.

Example

```
MediusServerEndGameResponse myEndGameResponse;
myEndGameResponse.Confirmation = MGCL_SUCCESS
```

See also

[MGCLEndGameResponse\(\)](#)

MediusServerJoinGameRequest

Information about a user that is trying to join a game on this host.

Link to file	Include file	Introduced	Last modified
librtmgcl.a	mgcl.h	1.0	February 1, 2002

Structure

```

typedef struct {
    char MessageID[MGCL_MESSAGEID_MAXLEN];           Message ID used for asynchronous request
                                                       processing.
    NetConnectionInfo ConnectInfo;                    The user's IP address.
} MediusServerJoinGameRequest;

```

Description

This structure is used in an unsolicited callback when a user requests to join a game on this game server host.

Notes

Contains the requesting client's connection information.

Example

N/A

See also

[MGCLServerJoinGameCallback](#)

MediusServerJoinGameResponse

Response to grant or deny a client request to join a game on this host.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	February 1, 2002

Structure

```
typedef struct {
    char MessageID[MGCL_MESSAGEID_MAXLEN];
    char Confirmation;
    char AccessKey[MGCL_ACCESSKEY_MAXLEN];
    RSA_KEY pubKey;
    int DmeClientIndex;
} MediusServerJoinGameResponse;
```

Message ID used for asynchronous request processing.

MGCL_SUCCESS or other value to indicate an error.

The access key generated by this host to allow a client to connect.

The security key used for encryption for the incoming client.

The DME client index reserved for the incoming client.

Description

Response to grant or deny a client request to join a game on this host. This is used in [MGCLJoinGameResponse\(\)](#).

Notes

The confirmation maps to MGCL_ERROR_CODE. This is set to MGCL_SUCCESS to denote that client is allowed to join the game.

Example

```
MediusServerJoinGameResponse myJoinGameResponse;
randomMTSetCharArray(authentication_key, AUTHENTICATION_KEY_SIZE);
akey_size = strlen(authentication_key);
DmeServerSaveConnectKeys(WorldID, SessionKey, skey_size, authentication_key,
    akey_size+1, CONNECTION_TIMEOUT);
myJoinGameResponse.Confirmation = MGCL_SUCCESS
memcpy(&(myJoinGameResponse.pubKey), &(G_ServicePublicKey), sizeof(RSA_KEY));
strncpy(myJoinGameResponse.AccessKey, authentication_key,
    AUTHENTICATION_KEY_SIZE);
MGCLJoinGameResponse(&myJoinGameResponse);
```

See also

[MGCLJoinGameResponse\(\)](#)

MediusServerMoveGameWorldOnMeRequest

Request to move a game world from one host to this host using the [MGCLMoveGameWorldOnMe\(\)](#) function.

Link to file	Include file	Introduced	Last modified
librtmgcl.a	mgcl.h	1.0	February 1, 2002

Structure

```

typedef struct {
    char MessageID[MGCL_MESSAGEID_MAXLEN];
    int CurrentMediusWorldID;
    int NewGameWorldID;

    NetAddressList AddressList;

} MediusServerMoveGameWorldOnMeRequest;

```

Message ID used for asynchronous request processing.

The Medius game world ID.

The new Medius game world ID. It is atomic across hosts. The old game world can not accidentally be used to reference the game.

The game servers IP address and port. Both internal (NAT) and external global addresses are available.

Description

Request by an MGCL host to migrate a game to this host. This is used only by peer-to-peer hosts. The CurrentMediusWorldID refers to the MediusWorldID originally returned in the [MediusServerCreateGameOnMeResponse](#). The NewGameWorldID is typically 1. This refers to the physical DME WorldID, not the MediusWorldID. This has a value greater than 1 only if this MGCL host can simultaneously maintain multiple game instances (as with game servers).

Notes

This structure is used for peer-to-peer games, and not for client-server games. For a game world to move on to this host, this must be a host (i.e., pre-authenticated with a session and fully ready server) For peer-to-peer titles that support host migration, all game participants should set themselves up as game servers before attempting to join a game.

Example

```

// Set the callback for notification when the host changes in the
// NetConnect call
NetConnectInParams MyNetConnectInParams;
MyNetConnectInParams.pfPeerToPeerHostChangeCallback = myHostChangeCB;
NetConnect(&MyNetConnectInParams, ...

// Save your client index.
NetGetMyClientIndex(&G_MyClientIndex, G_MyDME_PEER-TO-PEER_ConnectionHandle);

// When the callback that the host change occurred, check to see if
// the host client index in the change data matches your save index.
// If so, then populate the MediusServerMoveGameWorldOnMeRequest structure
// and call MGCLMoveGameWorldOnMe()
myHostChangeCB(NetPeerToPeerHostChangeData *pHostChangeData) {
    if (G_MyClientIndex == pHostChangeData->HostClientIndex) {
        MGCLMoveGameWorldOnMe(...)
    }
}

```

See also

[MGCLMoveGameWorldOnMe\(\)](#)

MediusServerMoveGameWorldOnMeResponse

Response structure invoked through the callback. [MGCLServerMoveGameWorldOnMeCallback\(\)](#).

Link to file	Include file	Introduced	Last modified
librtmgcl.a	mgcl.h	1.0	February 1, 2002

Structure

```
typedef struct {
    char MessageID[MGCL_MESSAGEID_MAXLEN];    Message ID used for asynchronous request
                                                processing.
    char Confirmation;                        MGCL_SUCCESS or other value to indicate an
                                                error.
    int MediusWorldID;                       The game world id of the newly migrated world.
} MediusServerMoveGameWorldOnMeResponse;
```

Description

Response to MediusServerMoveGameWorldOnMe()

Notes

The confirmation maps to MGCL_ERROR_CODE.

Example

```
MGCLServerMoveGameWorldOnMeCallback()
(MediusServerMoveGameWorldOnMeResponse *ThisPacket, void* pUserData)
assert(ThisPacket->Confirmation == MGCL_SUCCESS);
G_MyMediusWorldID = ThisPacket->MediusWorldID;
```

See also

[MGCLServerMoveGameWorldOnMeCallback\(\)](#), [MGCLMoveGameWorldOnMe\(\)](#)

MediusServerReportType

This is the MGCL host/server report (total capacity and total state) used in [MGCLSendServerReport\(\)](#).

Link to file	Include file	Introduced	Last modified
librtmgcl.a	mgcl.h	1.0	February 1, 2002

Structure

typedef struct {	
char <i>SessionKey</i> [MGCL_SESSIONKEY_MAXLEN];	This is a server session key. It is needed because it first comes in on a redirected connection and can not look up the server by connection. MGCL populates it internally.
short <i>MaxWorlds</i> ;	Maximum number of game worlds supported by this game server.
short <i>MaxPlayersPerWorld</i> ;	The maximum number of players per game world.
short <i>ActiveWorldCount</i> ;	The number of active game worlds on this game server. Usually 1 for peer-to-peer hosts, or more for DME game servers.
short <i>TotalActivePlayers</i> ;	The total number of active players connected to this game server.
MGCL_ALERT_LEVEL <i>AlertLevel</i> ;	Alert level to allow for load balancing.
} MediusServerReportType;	

Description

This is the MGCL host/server report (total capacity and total state) used in [MGCLSendServerReport\(\)](#).

Notes

If a peer-to-peer host is sending this report, then MaxWorlds is set to "1" and MaxPlayersPerWorld should be set to the same value as MaxClients in NetHostPeerToPeerInParams.

Example

```

{
    MGCL_ERROR_CODE errorCode;
    MediusServerReportType myServerReport;

    myServerReport.SessionKey
    errorCode = MGCLSendServerReport (&myServerReport);
}
```

See also

[MGCLSendServerReport\(\)](#)

MediusServerSessionBeginRequest

Request to begin a new Server Session (first MGCL "send" call).

Link to file	Include file	Introduced	Last modified
librtmgcl.a	mgcl.h	1.0	February 1, 2002

Structure

```
typedef struct {
    char MessageID[MGCL_MESSAGEID_MAXLEN];
    int LocationID;
    int ApplicationID;

    MGCL_GAME_HOST_TYPE ServerType;

    char ServerVersion[MGCL_SERVERVERSION_MAXLEN];
    int Port;
} MediusServerSessionBeginRequest;
```

Message ID used for asynchronous request processing.

Location ID for this game server.

Application ID for this game server. Set to the title's application ID, or zero for a generic DME client-server game server.

Game server hosting type. PEER-TO-PEER, Integrated, Client-Server, etc...

Server version internally populated.

Port internally populated.

Description

Structure used to start a MGCL session via [MGCLSessionBegin\(\)](#)

Notes

This is the first call made after connecting to an authentication server.

Example

```
{
    MediusServerSessionBeginRequest    stMGCLSessionBeginRequest;
    memset(&stMGCLSessionBeginRequest,
        0,
        sizeof(MediusServerSessionBeginRequest));

    strcpy(stMGCLSessionBeginRequest.MessageID, "1");
    stMGCLSessionBeginRequest.LocationID    = 0;
    stMGCLSessionBeginRequest.ApplicationID = G_iMyApplicationID;
    stMGCLSessionBeginRequest.ServerType   = MGCLGameHostPeerToPeer;
    // The ServerVersion and Port are handled internally.
}
```

See also

[MGCLSessionBegin\(\)](#)

MediusServerSessionBeginResponse

Structure used in the response callback to MediusServerSessionBegin().

Link to file	Include file	Introduced	Last modified
librtmgcl.a	mgcl.h	1.0	February 1, 2002

Structure

```

typedef struct {
    char MessageID[MGCL_MESSAGEID_MAXLEN];           Message ID used for asynchronous request
                                                       processing.
    char Confirmation;                                MGCL_SUCCESS or other value to indicate an
                                                       error.
    NetConnectionInfo ConnectInfo;                    NAT (Network Address Translation) service IP/Port
                                                       session key. This is persisted internally in MGCL
} MediusServerSessionBeginResponse;
    
```

Description

Structure used in the response callback to MediusServerSessionBegin()

Notes

This is the response from the first call made after connecting to an authentication server. The confirmation maps to MGCL_ERROR_CODE.

Example

```

void MyMGCLServerSessionBeginCallback (
    MediusServerSessionBeginResponse*  pResponse,
    void* arg)
{
    if (pResponse->Confirmation == MGCL_SUCCESS)
    {
        {
            // Save the MGCL server session key. Do not confuse this with
            // the Medius Client session key. This session key is for the host
            // as a game server, and not as a Medius Client.
            memcpy(G_MyMGCLSessionKey, pResponse->ConnectInfo.aSessionKey,
                sizeof(pResponse->ConnectInfo.aSessionKey));
        }
    }
    else
    {
        {
            printf("Error with session begin\n");
        }
    }
}
    
```

See also

[MGCLSessionBegin\(\)](#)

MediusServerSessionEndRequest

Request to end an MGCL Session (graceful "leave").

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	February 1, 2002

Structure

```
typedef struct {
    char MessageID[MGCL_MESSAGEID_MAXLEN];    Message ID used for asynchronous request
                                                processing.
} MediusServerSessionEndRequest;
```

Description

Request to end an MGCL Session (graceful "leave")

Notes

This is the request made to close out a session.

Example

```
{
    MediusServerSessionEndRequest stEndRequest;

    memset(&stEndRequest, 0, sizeof(MediusServerSessionEndRequest));
    strcpy(stEndRequest.MessageID, "1");

    mgcl_result = MGCLSessionEnd( &stEndRequest,
                                  MyMGCLServerEndSessionCallback, NULL);
}
```

See also

[MGCLSessionEnd\(\)](#)

MediusServerSessionEndResponse

Structure used in the response callback to [MGCLSessionEnd\(\)](#).

Link to file	Include file	Introduced	Last modified
librtmgcl.a	mgcl.h	1.0	February 1, 2002

Structure

```
typedef struct {
    char MessageID[MGCL_MESSAGEID_MAXLEN];           Message ID used for asynchronous request
                                                       processing.
    char Confirmation;                                MGCL_SUCCESS or other value to indicate an
                                                       error.
} MediusServerSessionEndResponse;
```

Description

Structure used in the response callback to [MGCLSessionEnd\(\)](#).

Notes

This is the response from the [MGCLSessionEnd\(\)](#) to close out the session. The confirmation maps to MGCL_ERROR_CODE.

Example

```
void MyMGCLServerEndSessionCallback(
    MediusServerSessionBeginResponse* pResponse,
    void* arg)
{
    if(pResponse->Confirmation == MGCL_SUCCESS)
    {
        G_GameServerActive = 0;
    }
    else
    {
        printf("Error ending session\n");
    }
}
```

See also

[MGCLSessionEnd\(\)](#)

MediusServerSetAttributesRequest

Request the structure used when calling [MGCLSetServerAttributes\(\)](#) to set the game server attributes based on the bit field in MGCL_SERVER_ATTRIBUTES.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	2.10	February 1, 2005

Structure

```
typedef struct {
    char MessageID[MGCL_MESSAGEID_MAXLEN];           Message ID used for asynchronous request
                                                         processing.
    int Attributes;                                   MGCL_SERVER_ATTRIBUTES bit-wise OR'ed flag.
    NetAddress ListenServerAddress;                  IP address and port for the listen server.
} MediusServerSetAttributesRequest;
```

Description

Request the structure used when calling [MGCLSetServerAttributes\(\)](#) to set the game server attributes based on the bit field in MGCL_SERVER_ATTRIBUTES.

Notes

This structure determines if this is a rebroadcast or a spectator type of server. Usually, peer-to-peer clients do not need to make this call.

Example

```
MediusServerSetAttributesRequest myRequest;
myRequest.Attributes = MGCL_SERVER_ALLOW_REBROADCAST;
```

See also

[MGCLSetServerAttributes\(\)](#)

MediusServerSetAttributesResponse

Response structure for the request to set the game server attributes.

Link to file	Include file	Introduced	Last modified
librtmgcl.a	mgcl.h	2.10	February 1, 2005

Structure

```
typedef struct {
    char MessageID[MGCL_MESSAGEID_MAXLEN];    Message ID used for asynchronous request
                                                processing.
    char Confirmation;                         MGCL_SUCCESS or other value to indicate an
                                                error.
} MediusServerSetAttributesResponse;
```

Description

Response structure in the callback invoked by [MGCLSetServerAttributes\(\)](#).

Notes

The Confirmation value maps to the MGCL_ERROR_CODE.

Example

In the callback of type MGCLServerSetAttributesCallback, the following parameters are passed back:

```
(MediusServerSetAttributesResponse *ThisPacket, void* pUserData)
assert(ThisPacket->Confirmation==MGCL_SUCCESS);
```

See also

[MGCLSetServerAttributes\(\)](#), [MGCLServerSetAttributesCallback](#)

MediusServerWorldReportOnMe

Outgoing status or state information for a world running on this host.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	February 1, 2002

Structure

typedef struct {	
char <i>MessageID</i> [MGCL_MESSAGEID_MAXLEN];	Message ID used for asynchronous request processing.
char <i>GameName</i> [MGCL_GAMENAME_MAXLEN];	Name of the game world.
char <i>GameStats</i> [MGCL_GAMESTATS_MAXLEN];	Updated game statistics string.
char <i>GamePassword</i> [MGCL_GAMEPASSWORD_MAXLEN];	Game password string, NULL terminated.
int <i>ApplicationID</i> ;	Application ID of this game world.
int <i>MaxClients</i> ;	Maximum number of clients on this game world.
int <i>MinClients</i> ;	Minimum number of clients on this game world.
int <i>PlayerCount</i> ;	Current player count in this game.
int <i>GameLevel</i> ;	Game level, title dependent.
int <i>PlayerSkillLevel</i> ;	Player skill level, title dependent.
int <i>RulesSet</i> ;	Rule set, title dependent.
int <i>GenericField1</i> ;	GenericField, title dependent.
int <i>GenericField2</i> ;	GenericField, title dependent.
int <i>GenericField3</i> ;	GenericField, title dependent.
int <i>GenericField4</i> ;	GenericField, title dependent.
int <i>GenericField5</i> ;	GenericField, title dependent.
int <i>GenericField6</i> ;	GenericField, title dependent.
int <i>GenericField7</i> ;	GenericField, title dependent.
int <i>GenericField8</i> ;	GenericField, title dependent.
int <i>MediusWorldID</i> ;	Medius game world ID.
MGCL_WORLD_STATUS <i>WorldStatus</i> ;	Staging, active, closed.
} MediusServerWorldReportOnMe;	

Description

Structure used in the world report for a world on this host. Outgoing status or state information for a world running on this host.

Notes

This host must be authenticated as a game server.

Example

```
{
    MediusServerWorldReportOnMe myWorldReport;
    MGCL_ERROR_CODE result;

    strncpy(myWorldReport.MessageID, "1", MGCL_MESSAGEID_MAXLEN-1);
    myWorldReport.MessageID[MGCL_MESSAGEID_MAXLEN-1]='\0';

    strncpy(myWorldReport.GameName, G_GameName, MGCL_GAMENAME_MAXLEN-1);
    myWorldReport.GameName[MGCL_GAMENAME_MAXLEN-1]='\0';
}
```

```

strncpy(myWorldReport.GameStats, G_GameStats, MGCL_GAMESTATS_MAXLEN-1);
myWorldReport.GameStats[MGCL_GAMESTATS_MAXLEN-1]='\0';

strncpy(myWorldReport.GamePassword, G_GamePassword,
        MGCL_GAMEPASSWORD_MAXLEN-1);
myWorldReport.GamePassword[MGCL_GAMEPASSWORD_MAXLEN-1]='\0';

myWorldReport.ApplicationID = G_ApplicationID;
myWorldReport.MinClients = G_MinClients;
myWorldReport.MaxClients = G_MaxClients;
myWorldReport.GameLevel = G_GameLevel ;
myWorldReport.PlayerSkillLevel = G_PlayerSkillLevel;
myWorldReport.RulesSet = G_RulesSet;
myWorldReport.GenericField1 = G_GenericField1;
myWorldReport.GenericField2 = G_GenericField2;
myWorldReport.GenericField3 = G_GenericField3;
myWorldReport.GenericField4 = G_GenericField4;
myWorldReport.GenericField5 = G_GenericField5;
myWorldReport.GenericField6 = G_GenericField6;
myWorldReport.GenericField7 = G_GenericField7;
myWorldReport.GenericField8 = G_GenericField8;

myWorldReport.MediusWorldID = G_MediusGameWorldID;
myWorldReport.WorldStatus = MGCL_WORLD_STAGING;
myWorldReport.PlayerCount = 0;

result = MGCLSendWorldReportOnMe(&myWorldReport);
if (result != MGCL_SUCCESS)
{
    printf("MGCLSendWorldReportOnMe failed");
}
}

```

See also

[MGCLSendWorldReportOnMe\(\)](#)

MediusServerWorldStatusRequest

Request from a Medius Server for the status of a particular world.

Link to file	Include file	Introduced	Last modified
librtmgcl.a	mgcl.h	1.0	February 1, 2002

Structure

```
typedef struct {
    char MessageID[MGCL_MESSAGEID_MAXLEN];    Message ID used for asynchronous request
                                                processing.
    int WorldID;                               Game world ID being requested.
} MediusServerWorldStatusRequest;
```

Description

Request from a Medius Server for the status of a particular world.

Notes

This is invoked through an unsolicited callback `MGCLServerWorldStatusCallback` as defined in [MGCLInitializeInParams](#).

Example

```
void MyWorldStatusCallback(
    MediusServerWorldStatusRequest *pRequest, void *arg)
{
    MediusServerWorldStatusResponse response;
    MGCL_ERROR_CODE errorCode;

    strncpy(response.MessageID, pRequest->MessageID, MGCL_MESSAGEID_MAXLEN);
    response.ApplicationID = G_MyApplicationId;
    response.MaxClients = G_MyMaxClients;
    response.ActiveClients = G_MyNumberActiveClients;
    response.Confirmation = MGCL_SUCCESS;

    errorCode = MGCLWorldStatusResponse(&response);
    if (errorCode != MGCL_SUCCESS)
    {
        printf("failed to return the response status\n");
    }
}
```

See also

[MGCLServerWorldStatusCallback](#)

MediusServerWorldStatusResponse

Response to the Medius Server for the status of a particular world.

Link to file	Include file	Introduced	Last modified
librtmgcl.a	mgcl.h	1.0	February 1, 2002

Structure

```
typedef struct {
    char MessageID[MGCL_MESSAGEID_MAXLEN];
    int ApplicationID;
    int MaxClients;
    int ActiveClients;
    char Confirmation;
} MediusServerWorldStatusResponse;
```

Message ID used for asynchronous request processing.

Application ID of the game world.

Maximum number of clients for the game world.

Number of active clients in the game world.

MGCL_SUCCESS or other value to indicate an error.

Description

Response to the Medius Server for the status of a particular world.

Notes

This is response sent as a result of the request by the unsolicited callback MGCLServerWorldStatusCallback as defined in [MGCLInitializeInParams](#).

Example

See [MediusServerWorldStatusRequest](#) for an example.

See also

[MGCLWorldStatusResponse\(\)](#)

MGCLConnectInParams

Input parameters to [MGCLConnect\(\)](#).

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	February 1, 2002

Structure

```
typedef struct {
    NetConnectionInfo ConnectionInfo;           Connection information for the host (Authentication
                                                or Proxy server)
    MGCLConnectCallback pfConnectCallback;      Callback to invoke when the connect succeeds or
                                                fails.
} MGCLConnectInParams;
```

Description

Request structure used when calling [MGCLConnect\(\)](#)

Notes

[MGCLSetDefaultConnectParams\(\)](#) should be used to set the default parameters before setting any of the fields. [MGCLSetAuthConnectParams\(\)](#) should be called when using this to connect to an authentication server. MGCL is either connecting to an authentication server, or to a proxy server.

Example

```
MGCLConnectInParams myMGCLConnectInParams;
MGCLSetDefaultConnectParams (&myMGCLConnectInParams);
```

See also

[MGCLSetDefaultConnectParams\(\)](#), [MGCLConnect\(\)](#), [MGCLSetAuthConnectParams\(\)](#)

MGCLConnectOutParams

Output parameters from [MGCLConnect\(\)](#).

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	February 1, 2002

Structure

```
typedef struct {  
    MGCL\_ERROR\_CODE ErrorCode;           Return code for the API call to MGCLConnect\(\).  
} MGCLConnectOutParams;
```

Description

Output structure returned when calling [MGCLConnect\(\)](#)

Notes

N/A

Example

```
MGCLConnectInParams myMGCLConnectInParams;  
MGCLConnectOutParams myMGCLConnectOutParams;  
// Set the input parameter values.  
MGCLConnect(&myMGCLConnectInParams, &myMGCLConnectOutParams);
```

See also

[MGCLConnect\(\)](#), [MGCL_ERROR_CODE](#)

MGCLCreateGameOnMeRequestInParams

Request structure to create a game on this host.

Link to file	Include file	Introduced	Last modified
librtmgcl.a	mgcl.h	1.0	February 1, 2005

Structure

```
typedef struct {
```

[MediusServerCreateGameOnMeRequest](#)

CreateGameRequest;

[MGCLServerCreateGameOnMeCallback](#)

CreateGameCallback;

HDME *ConnectionHandle;*

[MediusServerCreateGameOnMeRequest](#): Request message structure.

[MGCLServerCreateGameOnMeCallback](#): Request completion callback.

HDME: Connection Handle returned by the call to [NetHostPeerToPeer](#). This parameter must not be NULL if the application has specified that it is hosting [MGCLGameHostPeerToPeer](#).

```
} MGCLCreateGameOnMeRequestInParams;
```

Description

Request structure used when calling [MGCLCreateGameOnMe\(\)](#)

Notes

Requires the [ConnectionHandler](#) from [NetHostPeerToPeer](#)

Example

```
MGCLCreateGameOnMeRequestInParams myCreateGameOnMeRequestInParams;
myCreateGameOnMeRequestInParams.CreateGameRequest.MinClients = 2;
// Fill in other fields
.
myCreateGameOnMeRequestInParams.CreateGameCallback = myCreateGameOnMeCallback;
.
// Use the output parameter from NetHostPeerToPeer() here.
myCreateGameOnMeRequestInParams.ConnectionHandle =
    myNetHostPeerToPeerOutParams.ConnectionHandle;
MGCLCreateGameOnMe(&myCreateGameOnMeRequestInParams, NULL);
```

See also

[MGCLCreateGameOnMeRequest\(\)](#)

MGCLDisconnectParams

Request structure to disconnect from a connection.

Link to file	Include file	Introduced	Last modified
librtmgcl.a	mgcl.h	1.0	February 1, 2002

Structure

```
typedef struct {  
    HDME ConnectionHandle;           DME connection handle to the host (Authentication  
                                     or Proxy server).  
  
    NetDisconnectReason DisconnectReason; Reason for disconnect. This should be  
                                         NetDisconnectNormal in the typical case.  
  
    MGCLConnectCallback MyDisconnectCallback; Callback to invoke when the disconnect is finished  
                                             processing.  
} MGCLDisconnectParams;
```

Description

Request structure used when calling [MGCLDisconnect\(\)](#) to disconnect from a connection.

Notes

This structure should be initialized with default values via [MGCLSetDefaultDisconnectParams\(\)](#)

Example

```
MGCLDisconnectParams myDisconnectParams;  
MGCLSetDefaultDisconnectParams (&myDisconnectParams);  
myDisconnectParams.ConnectionHandle = MyDMEHandle;  
myDisconnectParams.DisconnectReason = NetDisconnectNormal;  
myDisconnectParams.MyDisconnectCallback = MyDisconnectCallback  
MGCLDisconnect (&myDisconnectParams, NULL);
```

See also

[MGCLDisconnect\(\)](#)

MGCLInitializeInParams

Input parameters to MGCLInitialize.

Link to file	Include file	Introduced	Last modified
librtmgcl.a	mgcl.h	1.0	October 31, 2004

Structure

typedef struct {	
RSA_KEYPAIR *pLocalKeyPair;	Local RSA Key pair.
DMEInitStatus Status;	Flag to denote whether or not the DME was initialized prior to the MGCLInitialize() call.
MGCLServerCreateGameCallback	Callback to be invoked when a game world is requested by Medius to be spawned on this host.
<i>MyServerCreateGameCallback;</i>	
void *MyServerCreateGameCallbackUserData;	User specific data for the create game callback.
MGCLServerCreateGameWithAttributesCallback	Callback to be invoked when a game world is request by Medius to be spawned on this host.
<i>MyServerCreateGameWithAttributesCallback;</i>	This game world creation request contains specific attributes.
void *MyServerCreateGameWithAttributesUserData;	User specific data for the create game with attributes callback.
MGCLServerJoinGameCallback	Callback to be invoked when a client requests to join a game on this host.
<i>MyServerJoinGameCallback;</i>	
void *MyServerJoinGameCallbackUserData;	User specific data for the join game callback.
MGCLServerEndGameCallback	Callback invoked when Medius requests that a game be terminated on this host.
<i>MyServerEndGameCallback;</i>	
void *MyServerEndGameCallbackUserData;	User specific data for the end game callback.
MGCLServerWorldStatusCallback	Callback to be invoked when Medius requests information on a particular game on this host.
<i>MyServerWorldStatusCallback;</i>	
void *MyServerWorldStatusCallbackUserData;	User specific data for the world status callback.
MGCLServerConnectGamesCallback	Callback to be invoked when a game server requests to connect to this host for the purposes of spectating. Not used for peer-to-peer hosts.
<i>MyServerConnectGamesCallback;</i>	
void *MyServerConnectGamesUserData;	User specific data for the connect games callback.
MGCLServerDisconnectPlayerCallback	Callback to be invoked when Medius requests that a particular player be disconnected from this host.
<i>MyServerDisconnectPlayerCallback;</i>	
void *MyServerDisconnectPlayerUserData;	User specific data for the disconnect player callback.
int MaxClientsPerConnection;	This value must be set to MaxPlayersPerChannel in mps.txt Represents MaxHostPerWorld on Proxy Server
int ApplicationID;	Application ID.
const RSA_KEYPAIR *pApplicationKeyPair;	Application-specific key pair, obtained from the software key library.
} MGCLInitializeInParams;	

Description

Request structure used when calling [MGCLInitialize\(\)](#)

Notes

This structure specifies all of the callbacks for any unsolicited message.

Example

```

MGCLInitializeInParams myMGCLInitializeInParams;
MGCLInitializeOutParams myMGCLInitializeOutParams;
RSA_KEYPAIR          softKeyPair;
KM_GetSoftwareKeyPair(&(softKeyPair.publicKey), &(softKeyPair.privateKey));
myMGCLInitializeInParams.pLocalKeyPair = &G_LocalKeyPair;
myMGCLInitializeInParams.Status = DMENotInitialized;
myMGCLInitializeInParams.MyServerCreateGameCallback = MyCreateGameCallback;
myMGCLInitializeInParams.MyServerCreateGameWithAttributesCallback =
    MyCreateGameWithAttributesCallback;
myMGCLInitializeInParams.MyServerJoinGameCallback = MyJoinGameCallback;
myMGCLInitializeInParams.MyServerEndGameCallback = MyEndGameCallback;
myMGCLInitializeInParams.MyServerWorldStatusCallback = MyWorldStatusCallback;
myMGCLInitializeInParams.MyServerConnectGamesCallback = MyConnectGamesCallback;
myMGCLInitializeInParams.MyServerDisconnectPlayerCallback =
    MyDisconnectPlayerCallback;
myMGCLInitializeInParams.MaxClientsPerConnection = MaxClientsPerWorld;
myMGCLInitializeInParams.ApplicationID = KM_GetSoftwareID();
myMGCLInitializeInParams.pApplicationKeyPair = &softKeyPair;
MGCLInitialize(&myMGCLInitializeInParams, &myMGCLInitializeOutParams);

```

See also

[MGCLInitialize\(\)](#)

MGCLInitializeOutParams

Output parameter from [MGCLInitialize\(\)](#).

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	February 1, 2002

Structure

```
typedef struct {
    MGCL\_ERROR\_CODE ErrorCode;           Error code.
} MGCLInitializeOutParams;
```

Description

Output parameter from [MGCLInitialize\(\)](#)

Notes

Contains the error code

Example

```
MGCLInitializeInParams myMGCLInitializeInParams;
MGCLInitializeOutParams myMGCLInitializeOutParams;
// Set myMGCLInitializeInParams
MGCLInitialize(&myMGCLInitializeInParams, &myMGCLInitializeOutParams);
```

See also

[MGCLInitialize\(\)](#)

This page intentionally left blank.

Chapter 4: Callback Functions

This page intentionally left blank.

MGCLConnectCallback

Callback invoked upon a connect event.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	October 1, 2001

Syntax

```
typedef void (*MGCLConnectCallback)(
    HDME ConnectionHandle,           DME Connection Handle to the Medius Proxy
                                     Server.
    MGCL_ERROR_CODE ConnectStatus,  MGCL_SUCCESS or
                                     MGCL_CONNECTION_FAILED
    void *pUserData);                Pointer to a response message data structure.
```

Description

Callback invoked upon a connect event.

Errors

None.

Notes

N/A

Return value

None.

Example

N/A

See also

Refer to mediustypes.h for definitions of response message data structures.

MGCLServerAuthenticationCallback

Callback invoked after the server authentication has completed.

Link to file	Include file	Introduced	Last modified
librtmgcl.a	mgcl.h	1.0	October 1, 2001

Syntax

```
typedef void (*MGCLServerAuthenticationCallback)(
    MediusServerAuthenticationResponse *ThisPacket,
    void *pUserData);
```

Pointer to a [MediusServerAuthenticationResponse](#) structure.

Pointer to a response message data structure.

Description

Callback invoked after the server authentication has completed.

Errors

None.

Notes

N/A

Return value

None.

Example

N/A

See also

Refer to mediustypes.h for definitions of response message data structures.

MGCLServerConnectGamesCallback

Unsolicited callback invoked when the Medius server requests a game world to connect to this game server. This is usually used for supporting spectator/broadcast worlds.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	October 1, 2001

Syntax

```
typedef void (*MGCLServerConnectGamesCallback)(
    MediusServerConnectGamesRequest *ThisPacket,    Pointer to a MediusServerConnectGamesRequest
                                                    structure.
    void *pUserData);                               Pointer to a response message data structure.
```

Description

Unsolicited callback invoked when the Medius server requests a game world to connect to this game server. This is usually used for supporting spectator/broadcast worlds.

Errors

None.

Notes

N/A

Return value

None.

Example

N/A

See also

Refer to mediustypes.h for definitions of response message data structures.

MGCLServerCreateGameCallback

Unsolicited callback invoked when a client is attempting to create a game on this server.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	October 1, 2001

Syntax

```
typedef void (*MGCLServerCreateGameCallback)(
    MediusServerCreateGameRequest *ThisPacket,      Pointer to a MediusServerCreateGameRequest
                                                    structure.
    void *pUserData);                               Pointer to a response message data structure.
```

Description

Unsolicited callback invoked when a client is attempting to create a game on this server.

Errors

None.

Notes

N/A

Return value

None.

Example

N/A

See also

Refer to mediustypes.h for definitions of response message data structures.

MGCLServerCreateGameOnMeCallback

Callback invoked upon a response from the server after a request to create a game on this host using [MGCLCreateGameOnMeRequest\(\)](#).

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	October 1, 2001

Syntax

```
typedef void (*MGCLServerCreateGameOnMeCallback)(
    MediusServerCreateGameOnMeResponse *ThisPacket,
    void *pUserData);
```

Pointer to a [MediusServerCreateGameOnMeResponse](#) structure.

Pointer to a response message data structure.

Description

Callback invoked upon a response from the server after a request to create a game on this host using [MGCLCreateGameOnMeRequest\(\)](#).

Errors

None.

Notes

N/A

Return value

None.

Example

N/A

See also

Refer to [mediustypes.h](#) for definitions of response message data structures.

MGCLServerCreateGameWithAttributesCallback

Unsolicited callback invoked when a client is attempting to create a game with attributes on this server.

Link to file	Include file	Introduced	Last modified
librtmgcl.a	mgcl.h	1.0	October 1, 2001

Syntax

```
typedef void
(*MGCLServerCreateGameWithAttributesCallback)(
    MediusServerCreateGameWithAttributesRequest
    *ThisPacket,
    void *pUserData);
```

Pointer to a [MediusServerCreateGameWithAttributesRequest](#) structure.

Pointer to a response message data structure.

Description

Unsolicited callback invoked when a client is attempting to create a game with attributes on this server.

Errors

None.

Notes

N/A

Return value

None.

Example

N/A

See also

Refer to mediustypes.h for definitions of response message data structures.

MGCLServerDisconnectPlayerCallback

Unsolicited callback invoked when the Medius server requests that a particular player be disconnected from this host.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	October 1, 2001

Syntax

```
typedef void (*MGCLServerDisconnectPlayerCallback)(
    MediusServerDisconnectPlayerRequest *ThisPacket,    Pointer to a
                                                         MediusServerDisconnectPlayerRequest structure.
    void *pUserData);                                  Pointer to a response message data structure.
```

Description

Unsolicited callback invoked when the Medius server requests that a particular player be disconnected from this host.

Errors

None.

Notes

N/A

Return value

None.

Example

N/A

See also

Refer to mediustypes.h for definitions of response message data structures.

MGCLServerEndGameCallback

Unsolicited callback invoked when Medius is telling this server to end a game (either nicely or brutally killing all connections).

Link to file	Include file	Introduced	Last modified
librtmgcl.a	mgcl.h	1.0	October 1, 2001

Syntax

```
typedef void (*MGCLServerEndGameCallback)(
    MediusServerEndGameRequest *ThisPacket,           Pointer to a MediusServerEndGameRequest
                                                       structure.
    void *pUserData);                                Pointer to a response message data structure.
```

Description

Unsolicited callback invoked when Medius is telling this server to end a game (either nicely or brutally killing all connections).

Errors

None.

Notes

N/A

Return value

None.

Example

N/A

See also

Refer to mediustypes.h for definitions of response message data structures.

MGCLServerEndGameOnMeCallback

Callback invoked upon a response from the server after a request to end a game on this host using [MGCLEndGameOnMeRequest\(\)](#).

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	October 1, 2001

Syntax

```
typedef void (*MGCLServerEndGameOnMeCallback)(
    MediusServerEndGameOnMeResponse *ThisPacket,    Pointer to a
                                                    MediusServerEndGameOnMeResponse structure.
    void *pUserData);                               Pointer to a response message data structure.
```

Description

Callback invoked upon a response from the server after a request to end a game on this host using [MGCLEndGameOnMeRequest\(\)](#).

Errors

None.

Notes

N/A

Return value

None.

Example

N/A

See also

Refer to [mediustypes.h](#) for definitions of response message data structures.

MGCLServerJoinGameCallback

Unsolicited callback invoked when a client is attempting to join a game on this server.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	October 1, 2001

Syntax

```
typedef int (*MGCLServerJoinGameCallback)(
    MediusServerJoinGameRequest *ThisPacket,      Pointer to a MediusServerJoinGameRequest
                                                    structure.
    void *pUserData);                             Pointer to a response message data structure.
```

Description

Unsolicited callback invoked when a client is attempting to join a game on this server.

Errors

None.

Notes

N/A

Return value

None.

Example

N/A

See also

Refer to mediustypes.h for definitions of response message data structures.

MGCLServerMoveGameWorldOnMeCallback

Callback invoked upon a response from the server after a request to migrate a game world from one server to this server using [MGCLMoveGameWorldOnMe\(\)](#). This is also referred to as host migration.

Link to file	Include file	Introduced	Last modified
librtmgcl.a	mgcl.h	1.0	October 1, 2001

Syntax

```
typedef void
(*MGCLServerMoveGameWorldOnMeCallback)(
    MediusServerMoveGameWorldOnMeResponse    Pointer to a
    *ThisPacket,                             MediusServerMoveGameWorldOnMeResponse
                                           structure.
    void *pUserData);                       Pointer to a response message data structure.
```

Description

Callback invoked upon a response from the server after a request to migrate a game world from one server to this server using [MGCLMoveGameWorldOnMe\(\)](#). This is also referred to as host migration.

Errors

None.

Notes

N/A

Return value

None.

Example

N/A

See also

Refer to mediustypes.h for definitions of response message data structures.

MGCLServerSessionBeginCallback

Callback invoked after the server has established a session.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	October 1, 2001

Syntax

```
typedef void (*MGCLServerSessionBeginCallback)(
    MediusServerSessionBeginResponse *ThisPacket,    Pointer to a MediusServerSessionBeginResponse
                                                    structure.
    void *pUserData);                                Pointer to a response message data structure.
```

Description

Callback invoked after the server has established a session.

Errors

None.

Notes

N/A

Return value

None.

Example

N/A

See also

Refer to mediustypes.h for definitions of response message data structures.

MGCLServerSessionEndCallback

Callback invoked after the server's session has terminated.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	October 1, 2001

Syntax

```
typedef void (*MGCLServerSessionEndCallback)(
    MediusServerSessionEndResponse *ThisPacket,
    void *pUserData);
```

Pointer to a [MediusServerSessionEndResponse](#) structure.

Pointer to a response message data structure.

Description

Callback invoked after the server's session has terminated. the callback is MGCLServerSessionEndCallback.

Errors

None.

Notes

N/A

Return value

None.

Example

N/A

See also

Refer to mediustypes.h for definitions of the response message data structures.

MGCLServerSetAttributesCallback

Callback invoked after this servers attributes have been set.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	October 1, 2001

Syntax

```
typedef void (*MGCLServerSetAttributesCallback)(
    MediusServerSetAttributesResponse *ThisPacket,    Pointer to a MediusServerSetAttributesResponse
                                                    structure.
    void *pUserData);                                Pointer to a response message data structure.
```

Description

Callback invoked after this servers attributes have been set.

Errors

None.

Notes

N/A

Return value

None.

Example

N/A

See also

Refer to mediustypes.h for definitions of response message data structures.

MGCLServerWorldStatusCallback

Unsolicited callback invoked when Medius is asking this server to respond with the status of a particular game world.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	October 1, 2001

Syntax

```
typedef void (*MGCLServerWorldStatusCallback)(
    MediusServerWorldStatusRequest *ThisPacket,      Pointer to a MediusServerWorldStatusRequest
                                                    structure.
    void *pUserData);                               Pointer to a response message data structure.
```

Description

Unsolicited callback invoked when Medius is asking this server to respond with the status of a particular game world.

Errors

None.

Notes

N/A

Return value

None.

Example

N/A

See also

[MediusServerWorldStatusResponse](#), [MGCLWorldStatusResponse\(\)](#)

Refer to mediustypes.h for definitions of response message data structures.

This page intentionally left blank.

Chapter 5: Functions

This page intentionally left blank.

MGCLAuthenticationSend

Send an authentication to the Medius Authentication Server.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	November 1, 2002

Syntax

```
MGCL_ERROR_CODE MGCLAuthenticationSend(  

  MediusServerAuthenticationRequest  

  *MyServerAuthenticationRequest,  

  MGCLServerAuthenticationCallback  

  MyServerAuthenticationCallback,  

  void *pUserData);
```

Request a structure containing the necessary information to authenticate a host

Callback function to use when the response from the server is received

Pointer to UserData available when callback is triggered

Description

Send a request to an authentication server.

Errors

None.

Notes

This is called after the [MGCLInitialize\(\)](#), after connecting to the authentication server, and after establishing a session with [MGCLSessionBegin\(\)](#).

Return value

MGCL_UNSUCCESSFUL, MGCL_SUCCESS, MGCL_SEND_FAILED, MGCL_INVALID_ARG

Example

```
errorCode = MGCLAuthenticationSend(&myServerAuthenticationRequest,  

    myServerAuthenticationCallback, NULL);
```

See also

[MGCLSessionBegin\(\)](#)

MGCLClose

Close the MGCL engine.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	February 1, 2002

Syntax

```
MGCL_ERROR_CODE MGCLClose(  
    void );
```

Description

Close the MGCL engine.

Errors

None.

Notes

This is called to close down and release the resource associated with MGCL.

Return value

MGCL_UNSUCCESSFUL, MGCL_SUCCESS, MGCL_SHUTDOWN_ERROR

Example

```
errorCode = MGCLClose();
```

See also

[MGCLInitialize\(\)](#)

MGCLConnect

Connect to a Medius Server.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	February 1, 2002

Syntax

```
MGCL_ERROR_CODE MGCLConnect(
MGCLConnectInParams *MyConnectInParams,      Input parameters for the server connection.
MGCLConnectOutParams *MyConnectOutParams,    Output parameters for the server connection.
void *pUserData);                             Pointer to UserData that is available when the
                                                callback is triggered.
```

Description

Connect to a Medius Server.

Errors

None.

Notes

This is called to connect to an authentication server or a proxy server. [MGCLUpdate\(\)](#) must be called to trigger the connect callback.

Return value

MGCL_UNSUCCESSFUL, MGCL_SUCCESS, MGCL_CONNECTION_FAILED, MGCL_INVALID_ARG

Example

```
errorCode = MGCLConnect (&myConnectInParams, &myConnectOutParams, NULL);
```

See also

[MGCLInitialize\(\)](#), [MGCLUpdate\(\)](#)

MGCLConnectGamesResponse

Respond to a request to connect a spectator world to a game world.

Link to file	Include file	Introduced	Last modified
librtmgcl.a	mgcl.h	2.9	November 1, 2004

Syntax

MGCL_ERROR_CODE MGCLConnectGamesResponse(
 MediusServerConnectGamesResponse
 *MyServerConnectGamesResponse); Response to the request to connect a spectator
 world to this host.

Description

Send a response to the Medius server regarding a request to connect a spectator game world to this host.

Errors

An uninitialized MGCL, or invalid parameters will result in MGCL_UNSUCCESSFUL.

Notes

This function is fire-and-forget.

Return value

MGCL_UNSUCCESSFUL, MGCL_SUCCESS, MGCL_SEND_FAILED

Example

```
errorCode = MGCLConnectGamesResponse (&myServerConnectGamesResponse) ;
```

See also

[MGCL_EVENT_TYPE](#)

MGCLCreateGameOnMeRequest

Request to create a game on this host.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	February 1, 2002

Syntax

MGCL_ERROR_CODE

MGCLCreateGameOnMeRequest(

MGCLCreateGameOnMeRequestInParams *pInParams, Input parameters about the game to be created on this host.

void *pUserData); Pointer to UserData available when the callback is triggered.

Description

Request to create a game on this host.

Errors

None.

Notes

This host must be authenticated as a game server before making this request.

Return value

MGCL_UNSUCCESSFUL, MGCL_SUCCESS, MGCL_SEND_FAILED, MGCL_INVALID_ARG

Example

```
errorCode = MGCLCreateGameOnMeRequest (&myInParams, NULL);
```

See also

[MGCLCreateGameOnMeRequestInParams](#)

MGCLCreateGameResponse

Respond to a request to have a game created on this host.

Link to file	Include file	Introduced	Last modified
librtmgcl.a	mgcl.h	1.0	February 1, 2002

Syntax

MGCL_ERROR_CODE MGCLCreateGameResponse(
MediusServerCreateGameResponse
*MyServerCreateGameResponse);

Response to the request to create a game world on this host.

Description

Respond to a request to have a game created on this host.

Errors

None.

Notes

The host must be initialized and authenticated as a server before games can be created.

Return value

MGCL_SUCCESS, MGCL_SEND_FAILED

Example

```

errorCode = MGCLCreateGameResponse (&myServerCreateGameResponse);

```

See also

MediusServerCreateGameResponse

MGCLCreateGameWithAttributesResponse

Respond to a request to have a game created on this host with attributes.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	2.9	November 1, 2004

Syntax

[MGCL_ERROR_CODE](#)

MGCLCreateGameWithAttributesResponse(

[MediusServerCreateGameWithAttributesResponse](#)

**MyServerCreateGameWithAttributesResponse*);

Response to a request to create a game world on this host.

Description

Respond to a request to have a game created on this host with attributes.

Errors

None.

Notes

The host must be initialized and authenticated as a server before games can be created here.

Return value

MGCL_SUCCESS, MGCL_SEND_FAILED

Example

```
errorCode = MGCLCreateGameWithAttributesResponse (&myServerCreateGameResponse) ;
```

See also

[MediusServerCreateGameWithAttributesResponse](#)

MGCLDisconnect

Disconnect from a Medius Server.

Link to file	Include file	Introduced	Last modified
librtmgcl.a	mgcl.h	1.0	February 1, 2002

Syntax

```
MGCL_ERROR_CODE MGCLDisconnect(  
    MGCLDisconnectParams *MyMGCLDisconnectParams,    Input parameters from which the server will  
                                                       disconnect.  
    void *pUserData);                                Pointer to UserData available when the callback is  
                                                       triggered.
```

Description

Disconnect from a Medius Server.

Errors

None.

Notes

MGCLUpdate() must be called to trigger the disconnect callback.

Return value

MGCL_UNSUCCESSFUL, MGCL_SUCCESS, MGCL_INVALID_ARG, MGCL_DISCONNECT_FAILED

Example

```
errorCode = MGCLDisconnect(&myMGCLDisconnectParams, NULL);
```

See also

MGCLConnect()

MGCLEndGameOnMeRequest

Request to end a game on this host.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	February 1, 2002

Syntax

```
MGCL_ERROR_CODE MGCLEndGameOnMeRequest(  

  MediusServerEndGameOnMeRequest      Input parameters for the game world to terminate.  

  *MyServerEndGameOnMeRequest,  

  MGCLServerEndGameOnMeCallback      Callback to denote whether or not the request was  

  MyServerEndGameOnMeCallback,        successful.  

  void *pUserData);                  Pointer to UserData available when callback is  

                                     triggered.
```

Description

Request to end a game on this host.

Errors

None.

Notes

[MGCLUpdate\(\)](#) must be called to trigger the callback.

Return value

MGCL_UNSUCCESSFUL, MGCL_SUCCESS, MGCL_SEND_FAILED

Example

```
errorCode = MGCLEndGameOnMeRequest(&myServerEndGameOnMeRequest,  

    myServerEndGameOnMeCallback, NULL);
```

See also

[MediusServerEndGameOnMeRequest](#), [MGCLServerEndGameOnMeCallback](#)

MGCLEndGameResponse

Respond to a request (from the Medius server) to stop a game on this host.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	2.9	November 1, 2004

Syntax

MGCL_ERROR_CODE MGCLEndGameResponse(

MediusServerEndGameResponse

**MyServerEndGameResponse*);

Response notification to indicate that a game has been terminated on this host

Description

Respond to a request (from the Medius server) to stop a game on this host.

Errors

None.

Notes

This is usually used in the context of DME game servers, and not in peer-to-peer games.

Return value

MGCL_SUCCESS, MGCL_SEND_FAILED

Example

```
errorCode = MGCLEndGameResponse (&myServerEndGameResponse) ;
```

See also

[MediusServerEndGameResponse](#)

MGCLGetMGCLBuildTimeStamp

Get the MGCL libraries build time stamp.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	February 1, 2002

Syntax

[MGCL_ERROR_CODE](#)

MGCLGetMGCLBuildTimeStamp(

char *TimeStamp);

Variable to populate with the time stamp. It should be defined as a character array of 64 bytes.

Description

Get the MGCL libraries build time stamp.

Errors

None.

Notes

This gets the build time stamp of the library.

Return value

MGCL_SUCCESS, MGCL_INVALID_ARG

Example

```
char caTimeStamp[64];
errorCode = MGCLGetMGCLBuildTimpStamp(caTimeStamp);
```

See also

[MGCLGetMGCLVersion\(\)](#)

MGCLGetMGCLVersion

Get the MGCL version information.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	February 1, 2002

Syntax

[MGCL_ERROR_CODE](#)

MGCLGetMGCLVersion(

char *Version);

Variable to populate with the MGCL version. It should be defined as a character array of
MEDIUS_GAME_COMM_LIBRARY_VERSION_NUMBER_MAXLEN.

Description

Get the MGCL version information.

Errors

None.

Notes

This is used to get the build version of the library.

Return value

MGCL_SUCCESS, MGCL_INVALID_ARG

Example

```
char caVersion[MEDIUS_GAME_COMM_LIBRARY_VERSION_NUMBER_MAXLEN];
errorCode = MGCLGetMGCLBuildTimeStamp(caVersion);
```

See also

[MGCLGetMGCLBuildTimeStamp\(\)](#)

MGCLGetServerTimestamp

Get the timestamp of the Medius Server to which this host is currently connected.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	2.9	November 1, 2004

Syntax

MGCL_ERROR_CODE MGCLGetServerTimestamp(

int *NetTimestamp);

Variable to populate with the time stamp.

Description

Get the timestamp of the Medius Server to which this host is currently connected.

Errors

None.

Notes

Get the time stamp of the server to which this host is connected. The parameter is set to zero if there is an error.

Return value

MGCL_SUCCESS, MGCL_DME_ERROR

Example

```
int myNetTimeStamp;
errorCode = MGCLGetServerTimestamp(&myNetTimeStamp);
```

See also

N/A

MGCLInitialize

Startup the MGCL engine.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	February 1, 2002

Syntax

```

MGCL_ERROR_CODE MGCLInitialize(
    MGCLInitializeInParams *MyInitializeInParams,      Input parameters to start MGCL
    MGCLInitializeOutParams *MyInitializeOutParams);    Output parameters to start MGCL

```

Description

Startup the MGCL engine

Errors

None.

Notes

Double initialization is forbidden. [MGCLClose\(\)](#) must be called before reinitializing.

Return value

MGCL_CALL_MGCL_CLOSE_BEFORE_REINITIALIZING, MGCL_SUCCESS, MGCL_INITIALIZATION_FAILED

Example

```

errorCode = MGCLInitialize(&myInitializeInParams, &myInitializeOutParams);

```

See also

[MGCLClose\(\)](#)

MGCLIPStringToBinary

Convert a string IP address to binary format for reports.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	2.9	November 1, 2004

Syntax

MGCL_ERROR_CODE MGCLIPStringToBinary(

unsigned char *IpBinary*[4],

Binary representation of the IPv4 address to be populated.

const char *IpString*[16]);

String representation of the IP String.

Description

Convert a string IP address to binary format for reports.

Errors

None.

Notes

This converts an IPv4 address from a string to a compact binary format.

Return value

MGCL_INITIALIZATION_FAILED, MGCL_SUCCESS

Example

```
char caIPAddress[16]="66.35.205.121";
unsigned char IpBinary[4];
errorCode = MGCLIPStringToBinary(IpBinary, caIPAddress);
```

See also

N/A

MGCLJoinGameResponse

Respond to a request for a player to join a game on this host.

Link to file	Include file	Introduced	Last modified
librtmgcl.a	mgcl.h	1.0	February 1, 2002

Syntax

MGCL_ERROR_CODE

MGCLJoinGameResponse(
MediusServerJoinGameResponse
*MyServerJoinGameResponse);

Response message to allow or disallow the client
to join a game on this host.

Description

Respond to a request for a player to join a game on this host.

Errors

None.

Notes

MGCLServerJoinGameCallback (in the initialization structure) is the incoming request. This call is the outgoing response.

Return value

MGCL_UNSUCCESSFUL, MGCL_SUCCESS, MGCL_SEND_FAILED

Example

```
errorCode = MGCLJoinGameResponse (&myServerJoinGameResponse);
```

See also

MediusServerJoinGameResponse

MGCLMoveGameWorldOnMe

Request by an MGCL host to migrate a game to this host.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	February 1, 2002

Syntax

```
MGCL_ERROR_CODE MGCLMoveGameWorldOnMe(
  MediusServerMoveGameWorldOnMeRequest
  *MyServerMoveGameWorldOnMeRequest,
  MGCLServerMoveGameWorldOnMeCallback
  MyServerMoveGameWorldOnMeCallback,
  void *pUserData);
```

Input parameters for the game world to migrate onto this host.

Callback when the migration is done.

Pointer to UserData available when the callback is triggered.

Description

Request by an MGCL host to migrate a game to this host.

Errors

None.

Notes

This is used only by peer-to-peer hosts. Also known as world migration.

Return value

MGCL_INVALID_ARG, MGCL_UNSUCCESSFUL, MGCL_SUCCESS, MGCL_SEND_FAILED

Example

```
errorCode = MGCLMoveGameWorldOnMe (&myServerMoveGameWorldOnMeRequest,
                                     MyServerMoveGameWorldOnMeCallback, NULL);
```

See also

[MediusServerMoveGameWorldOnMeRequest](#)

MGCLSendConnectNotification

Notifies Medius that a player has connected or disconnected from this server.

Link to file	Include file	Introduced	Last modified
librtmgcl.a	mgcl.h	2.9	November 1, 2004

Syntax

MGCL_ERROR_CODE

MGCLSendConnectNotification(
MediusServerConnectNotificationType
*MyServerConnectNotification);

Request a structure containing the connect/disconnect, world ID and player's session key.

Description

Send a Connection event message that a client connected to or disconnected from a game world on this host.

Errors

Invalid parameters result in MGCL_UNSUCCESSFUL.

Notes

This function is fire-and-forget.

Return value

MGCL_UNSUCCESSFUL, MGCL_SUCCESS, MGCL_SEND_FAILED

Example

```
errorCode = MGCLSendConnectNotification(&myServerConnectNotification);
```

See also

[MGCL_EVENT_TYPE](#)

MGCLSendServerReport

Send a server report for this host.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	February 1, 2002

Syntax

```
MGCL_ERROR_CODE MGCLSendServerReport(  
  MediusServerReportType *MyServerReport);          Server report.
```

Description

Send a server report for this host.

Errors

None.

Notes

This is used as a heartbeat, and is a fire-and-forget message. Ensure that one message is sent every minute to keep this server alive. Failure to send a periodic report will result in the de-registration of this host as a game server. The host must then re-authenticate if it wants to be able to be a game server again.

Return value

MGCL_UNSUCCESSFUL, MGCL_SUCCESS, MGCL_SEND_FAILED

Example

```
errorCode = MGCLSendServerReport (&myServerReport);
```

See also

[MediusServerReportType](#)

MGCLSendWorldReportOnMe

Send a world status report for a game on this host.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	February 1, 2002

Syntax

```
MGCL_ERROR_CODE MGCLSendWorldReportOnMe(  
  MediusServerWorldReportOnMe                Game world report.  
  *MyWorldReportOnMe);
```

Description

Send a world status report for a game on this host.

Errors

None.

Notes

This is used as a heartbeat for the game world, and is a fire-and-forget message. Ensure that one message is sent every minute to keep the world alive.

Return value

MGCL_UNSUCCESSFUL, MGCL_SUCCESS, MGCL_SEND_FAILED

Example

```
errorCode = MGCLSendWorldReportOnMe(&myWorldReportOnMe);
```

See also

[MediusServerWorldReportOnMe](#)

MGCLSessionBegin

Begin a ServerSession with the Medius Authentication Server.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	February 1, 2002

Syntax

```
MGCL_ERROR_CODE MGCLSessionBegin(  

  MediusServerSessionBeginRequest           Input parameters to start the session.  

  *MyServerSessionBeginRequest,  

  MGCLServerSessionBeginCallback           Callback to be invoked when the session is started.  

  MyServerSessionBeginCallback,  

  void *pUserData);                          Pointer to UserData that is available when the  

                                              callback is triggered.
```

Description

Begin a ServerSession with the Medius Authentication Server.

Errors

None.

Notes

MGCL must be initialized before using MGCLSessionBegin.

Return value

MGCL_UNSUCCESSFUL, MGCL_SUCCESS, MGCL_SEND_FAILED

Example

```
errorCode = MGCLSessionBegin(&myServerSessionBeginRequest,  

                             mySessionBeginCallback, NULL);
```

See also

[MediusServerSessionBeginRequest](#)

MGCLSessionEnd

End a ServerSession with the Medius Authentication Server.

Link to file	Include file	Introduced	Last modified
librtmgcl.a	mgcl.h	1.0	February 1, 2002

Syntax

MGCL_ERROR_CODE MGCLSessionEnd(MediusServerSessionEndRequest <i>*MyServerSessionEndRequest</i> , MGCLServerSessionEndCallback <i>MyServerSessionEndCallback</i> , void *pUserData);	<p>Input parameters to end the session.</p> <p>Callback to be invoked when the session is terminated</p> <p>Pointer to UserData available when callback is triggered.</p>
--	---

Description

End a ServerSession with the Medius Authentication Server.

Errors

None.

Notes

MGCL must be initialized before using MGCLSessionBegin.

Return value

MGCL_UNSUCCESSFUL, MGCL_SUCCESS, MGCL_SEND_FAILED

Example

```
errorCode = MGCLSessionEnd(&myServerSessionEndRequest,  
                           mySessionEndCallback, NULL);
```

See also

[MediusServerSessionEndRequest](#)

MGCLSetAuthConnectParams

Sets the default parameters to connect to an authentication server.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	February 1, 2002

Syntax

```
MGCL_ERROR_CODE MGCLSetAuthConnectParams(
MGCLConnectInParams *pConnectInParams,      Parameter to be populated
const char *szIPAddress,                    IP Address of the authentication server
const int iPort,                             Port of the authentication server
const int iWorldID,                          WorldID of the authentication server
MGCLConnectCallback pfConnectCallback);      The connect callback to be used after connecting
                                              to the authentication server
```

Description

After MGCLSetDefaultConnectParams is called, the same structure should be passed into this function to prepare for connection to a Medius Authentication Server. This same structure can then be used to call [MGCLConnect\(\)](#).

Errors

None.

Notes

A NULL pConnectInParams results in a MGCL_INVALID_ARG. pfConnectCallback should always be defined

Return value

MGCL_INVALID_ARG, MGCL_SUCCESS

Example

```
errorCode = MGCLSetAuthConnectParams(&myConnectInParams, "66.35.205.121",
                                     10075, 1, myConnectCallback);
```

See also

[MGCLConnect\(\)](#)

MGCLSetDefaultConnectParams

Initialize a new [MGCLConnectInParams](#) structure.

Link to file	Include file	Introduced	Last modified
librtmgcl.a	mgcl.h	1.0	February 1, 2002

Syntax

MGCL_ERROR_CODE

```

MGCLSetDefaultConnectParams(
    MGCLConnectInParams *pConnectInParams);           Parameters to set with default values.

```

Description

Initialize a new [MGCLConnectInParams](#) structure.

Errors

None.

Notes

This should be called after [MGCLInitialize\(\)](#). This same structure must be passed into [MGCLSetAuthConnectParams](#).

Return value

MGCL_INVALID_ARG, MGCL_SUCCESS

Example

```

errorCode = MGCLSetDefaultConnectParams(&myConnectInParams);

```

See also

[MGCLConnectInParams](#)

MGCLSetDefaultDisconnectParams

Initializes the [MGCLDisconnectParams](#) structure with a reasonable set of default values.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	2.9	November 1, 2004

Syntax

[MGCL_ERROR_CODE](#)

MGCLSetDefaultDisconnectParams(

[MGCLDisconnectParams](#) *MyMGCLDisconnectParams); Input parameters to set to a reasonable value.

Description

Initializes the [MGCLDisconnectParams](#) structure with a reasonable set of default values.

Errors

None.

Notes

A NULL parameter results in MGCL_INVALID_ARG.

Return value

MGCL_SUCCESS, MGCL_INVALID_ARG

Example

```
errorCode = MGCLSetDefaultDisconnectParams (&myMGCLDisconnectParams);
```

See also

[MGCLDisconnect\(\)](#)

MGCLSetServerAttributes

Set the attributes for this host.

Link to file	Include file	Introduced	Last modified
librtmgcl.a	mgcl.h	2.9	November 1, 2004

Syntax

MGCL_ERROR_CODE MGCLSetServerAttributes(MediusServerSetAttributesRequest <i>*MyServerSetAttributesRequest</i> ,	Request a structure containing the information required to set the attributes (as a normal or spectator server)
MGCLServerSetAttributesCallback <i>MyServerSetAttributesCallback</i> ,	Callback function to use when the response from the authentication server is received
void *pUserData);	Pointer to UserData available when callback is triggered

Description

Send a request to set server attributes.

Errors

None.

Notes

This is called after the host has authenticated the authentication server, and is still connected to the authentication server. Peer-to-peer clients usually do not need to make this call.

Return value

MGCL_UNSUCCESSFUL, MGCL_SUCCESS, MGCL_SEND_FAILED

Example

```
errorCode = MGCLSetServerAttributes(&myServerSetAttributesRequest,  
myServerSetAttributesCallback, NULL);
```

See also

[MGCLSessionBegin\(\)](#)

MGCLUpdate

Initiates any incoming or outgoing network traffic.

<i>Link to file</i>	<i>Include file</i>	<i>Introduced</i>	<i>Last modified</i>
librtmgcl.a	mgcl.h	1.0	November 1, 2002

Syntax

```
MGCL_ERROR_CODE MGCLUpdate(
void );
```

Description

Initiates any incoming or outgoing network traffic.

Errors

None.

Notes

After MGCL is initialized, this should be called every frame and must be called on a periodic basis (every 1 second or less). All data in the outbound buffers are transferred across the network when this function is called.

Return value

MGCL_UNSUCCESSFUL, MGCL_SUCCESS, MGCL_NETWORK_ERROR, MGCL_SEND_FAILED, MGCL_NATRESOLVE_FAILED

Example

```
while(mainLoop) {
    // Render;
    // Various MGCL calls and reports
    errorCode = MGCLUpdate();
}
```

See also

N/A

MGCLWorldStatusResponse

Respond to a request from the medius server about information on a particular world on this host.

Link to file	Include file	Introduced	Last modified
librtmgcl.a	mgcl.h	2.9	November 1, 2004

Syntax

MGCL_ERROR_CODE MGCLWorldStatusResponse(
MediusServerWorldStatusResponse
*MyServerWorldStatusResponse);

Response to a request for information about a world on this host.

Description

Send a response back to the Medius server regarding a request for the status of a game on this host.

Errors

An uninitialized MGCL or invalid parameters will result in MGCL_UNSUCCESSFUL.

Notes

This function is fire-and-forget.

Return value

MGCL_UNSUCCESSFUL, MGCL_SUCCESS, MGCL_SEND_FAILED

Example

```

errorCode = MGCLWorldStatusResponse(&myServerWorldStatusResponse);

```

See also

MediusServerWorldStatusResponse

Index

This page intentionally left blank.

Index

A

AccessKey.....3-21
 AccountID.....3-8
 ActiveClients.....3-34
 ActiveWorldCount.....3-24
 AddressList.....3-3, 3-8, 3-22
 AlertLevel.....3-24
 ApplicationID3-8, 3-11, 3-13, 3-25, 3-31, 3-34, 3-39
 Attributes3-13, 3-29

B

BrutalFlag3-18

C

Confirmation3-4, 3-6, 3-10, 3-12, 3-14, 3-17, 3-19, 3-21, 3-23, 3-26, 3-28, 3-30, 3-34
 ConnectEventType3-7
 ConnectInfo3-4, 3-20, 3-26
 ConnectionHandle3-37, 3-38
 ConnectionInfo3-35
 CreateGameCallback.....3-37
 CreateGameRequest3-37
 CurrentMediusWorldID.....3-22

D

DisconnectReason.....3-38
 DmeClientIndex3-15, 3-21
 DMEInitialized2-3
 DMEInitStatus2-3
 DMENotInitialized.....2-3
 DmeWorldID3-15

E

ErrorCode.....3-36, 3-41
 ExtraDMEInitStatus.....2-3
 ExtraMGCLAlertCode2-4
 ExtraMGCLGameHost2-8
 ExtraMGCLStatusCode2-5
 ExtraMGCLTrustLevelCode2-10
 ExtraMGCLWorldStatusCode2-12

G

GameHostType3-8
 GameLevel3-8, 3-31
 GameName3-8, 3-31
 GamePassword.....3-8, 3-31
 GameStats3-8, 3-31
 GameWorldID3-5, 3-6

GenericField13-8, 3-31
 GenericField23-8, 3-31
 GenericField33-8, 3-31
 GenericField43-8, 3-31
 GenericField53-8, 3-31
 GenericField63-8, 3-31
 GenericField73-8, 3-31
 GenericField83-8, 3-31

L

ListenServerAddress3-29
 LocationID3-25

M

MAX_CLIENTS_PER_WORLD1-3
 MaxClients3-8, 3-11, 3-13, 3-31, 3-34
 MaxClientsPerConnection3-39
 MaxPlayersPerWorld3-24
 MaxWorlds3-24
 MEDIUS_GAME_COMM_LIBRARY_VERSION_NUMBER_M
 AXLEN1-4
 MediusServerAuthenticationRequest3-3
 MediusServerAuthenticationResponse.....3-4
 MediusServerConnectGamesRequest3-5
 MediusServerConnectGamesResponse3-6
 MediusServerConnectNotificationType3-7
 MediusServerCreateGameOnMeRequest3-8
 MediusServerCreateGameOnMeResponse3-10
 MediusServerCreateGameRequest.....3-11
 MediusServerCreateGameResponse.....3-12
 MediusServerCreateGameWithAttributesRequest.....3-13
 MediusServerCreateGameWithAttributesResponse3-14
 MediusServerDisconnectPlayerRequest3-15
 MediusServerEndGameOnMeRequest3-16
 MediusServerEndGameOnMeResponse.....3-17
 MediusServerEndGameRequest.....3-18
 MediusServerEndGameResponse3-19
 MediusServerJoinGameRequest.....3-20
 MediusServerJoinGameResponse3-21
 MediusServerMoveGameWorldOnMeRequest3-22
 MediusServerMoveGameWorldOnMeResponse3-23
 MediusServerReportType3-24
 MediusServerSessionBeginRequest3-25
 MediusServerSessionBeginResponse.....3-26
 MediusServerSessionEndRequest3-27
 MediusServerSessionEndResponse3-28
 MediusServerSetAttributesRequest3-29
 MediusServerSetAttributesResponse.....3-30
 MediusServerWorldReportOnMe3-31
 MediusServerWorldStatusRequest3-33

MediusServerWorldStatusResponse	3-34	MGCL_UNSUCCESSFUL	2-5
MediusWorldID	3-10, 3-16, 3-23, 3-31	MGCL_VERSION_BUILD	1-14
MediusWorldUID	3-7, 3-13	MGCL_VERSION_MAJOR	1-15
MessageID .. 3-3, 3-4, 3-5, 3-6, 3-8, 3-10, 3-11, 3-12, 3-13, 3-14, 3-16, 3-17, 3-18, 3-19, 3-20, 3-21, 3-22, 3-23, 3- 25, 3-26, 3-27, 3-28, 3-29, 3-30, 3-31, 3-33, 3-34		MGCL_VERSION_MINOR	1-16
MGCL_ACCESSKEY_MAXLEN	1-5	MGCL_WORLD_ACTIVE	2-12
MGCL_ALERT_HIGH	2-4	MGCL_WORLD_ALLOW_SPECTATORS	2-11
MGCL_ALERT_LEVEL	2-4	MGCL_WORLD_ATTRIBUTES	2-11
MGCL_ALERT_LOW	2-4	MGCL_WORLD_CLOSED	2-12
MGCL_ALERT_MID	2-4	MGCL_WORLD_EXTRA	2-11
MGCL_ALERT_NONE	2-4	MGCL_WORLD_INACTIVE	2-12
MGCL_AUTHENTICATION_FAILED	2-5	MGCL_WORLD_INTERNAL	2-11
MGCL_CALL_MGCL_CLOSE_BEFORE_REINITIALIZING	2-5	MGCL_WORLD_NONE	2-11
MGCL_CONNECTION_ERROR	2-5	MGCL_WORLD_STAGING	2-12
MGCL_CONNECTION_FAILED	2-5	MGCL_WORLD_STATUS	2-12
MGCL_DISCONNECT_FAILED	2-5	MGCL_WORLD_TYPE_INFORMER	2-11
MGCL_DME_ERROR	2-5	MGCL_WORLD_TYPE_SPECTATOR	2-11
MGCL_ERROR_CODE	2-5	MGCL_WORLDID_INUSE	2-5
MGCL_EVENT_CLIENT_CONNECT	2-7	MGCLAuthenticationSend	5-3
MGCL_EVENT_CLIENT_DISCONNECT	2-7	MGCLClose	5-4
MGCL_EVENT_EXTRA	2-7	MGCLConnect	5-5
MGCL_EVENT_TYPE	2-7	MGCLConnectCallback	4-3
MGCL_GAME_HOST_TYPE	2-8	MGCLConnectGamesResponse	5-6
MGCL_GAME_NAME_EXISTS	2-5	MGCLConnectInParams	3-35
MGCL_GAMENAME_MAXLEN	1-6	MGCLConnectOutParams	3-36
MGCL_GAMEPASSWORD_MAXLEN	1-7	MGCLCreateGameOnMeRequest	5-7
MGCL_GAMESTATS_MAXLEN	1-8	MGCLCreateGameOnMeRequestInParams	3-37
MGCL_INITIALIZATION_FAILED	2-5	MGCLCreateGameResponse	5-8
MGCL_INVALID_ARG	2-5	MGCLCreateGameWithAttributesResponse	5-9
MGCL_MESSAGEID_MAXLEN	1-9	MGCLDisconnect	5-10
MGCL_NATRESOLVE_FAILED	2-5	MGCLDisconnectParams	3-38
MGCL_NETWORK_ERROR	2-5	MGCLEndGameOnMeRequest	5-11
MGCL_NOT_CONNECTED	2-5	MGCLEndGameResponse	5-12
MGCL_NOT_TRUSTED	2-10	MGCLGameHostClientServer	2-8
MGCL_NUM_GAME_WORLDS_PER_LOBBY_WORLD_EXC EDED	2-5	MGCLGameHostClientServerAuxUDP	2-8
MGCL_SEND_FAILED	2-5	MGCLGameHostIntegratedServer	2-8
MGCL_SERVER_ALLOW_INFORMER	2-9	MGCLGameHostLANPlay	2-8
MGCL_SERVER_ALLOW_MONITOR	2-9	MGCLGameHostPeerToPeer	2-8
MGCL_SERVER_ALLOW_REBROADCAST	2-9	MGCLGetMGCLBuildTimeStamp	5-13
MGCL_SERVER_ALLOW_SPECTATOR	2-9	MGCLGetMGCLVersion	5-14
MGCL_SERVER_ATTRIBUTES	2-9	MGCLGetServerTimeStamp	5-15
MGCL_SERVER_EXTRA	2-9	MGCLInitialize	5-16
MGCL_SERVER_NONE	2-9	MGCLInitializeInParams	3-39
MGCL_SERVERIP_MAXLEN	1-10	MGCLInitializeOutParams	3-41
MGCL_SERVERPORT_MAXLEN	1-11	MGCLIPStringToBinary	5-17
MGCL_SERVERVERSION_MAXLEN	1-12	MGCLJoinGameResponse	5-18
MGCL_SESSIONBEGIN_FAILED	2-5	MGCLMoveGameWorldOnMe	5-19
MGCL_SESSIONEND_FAILED	2-5	MGCLSendConnectNotification	5-20
MGCL_SESSIONKEY_MAXLEN	1-13	MGCLSendServerReport	5-21
MGCL_SHUTDOWN_ERROR	2-5	MGCLSendWorldReportOnMe	5-22
MGCL_SUCCESS	2-5	MGCLServerAuthenticationCallback	4-4
MGCL_TRUST_LEVEL	2-10	MGCLServerConnectGamesCallback	4-5
MGCL_TRUSTED	2-10	MGCLServerCreateGameCallback	4-6
		MGCLServerCreateGameOnMeCallback	4-7
		MGCLServerCreateGameWithAttributesCallback	4-8
		MGCLServerDisconnectPlayerCallback	4-9

MGCLServerEndGameCallback	4-10
MGCLServerEndGameOnMeCallback	4-11
MGCLServerJoinGameCallback	4-12
MGCLServerMoveGameWorldOnMeCallback	4-13
MGCLServerSessionBeginCallback	4-14
MGCLServerSessionEndCallback	4-15
MGCLServerSetAttributesCallback	4-16
MGCLServerWorldStatusCallback	4-17
MGCLSessionBegin	5-23
MGCLSessionEnd	5-24
MGCLSetAuthConnectParams	5-25
MGCLSetDefaultConnectParams	5-26
MGCLSetDefaultDisconnectParams	5-27
MGCLSetServerAttributes	5-28
MGCLUpdate	5-29
MGCLWorldStatusResponse	5-30
MinClients	3-8, 3-31
MyDisconnectCallback	3-38
MyServerConnectGamesCallback	3-39
MyServerConnectGamesUserData	3-39
MyServerCreateGameCallback	3-39
MyServerCreateGameCallbackUserData	3-39
MyServerCreateGameWithAttributesCallback	3-39
MyServerCreateGameWithAttributesUserData	3-39
MyServerDisconnectPlayerCallback	3-39
MyServerDisconnectPlayerUserData	3-39
MyServerEndGameCallback	3-39
MyServerEndGameCallbackUserData	3-39
MyServerJoinGameCallback	3-39
MyServerJoinGameCallbackUserData	3-39
MyServerWorldStatusCallback	3-39
MyServerWorldStatusCallbackUserData	3-39

N

NewGameWorldID	3-22
----------------------	------

P

pApplicationKeyPair	3-39
pfConnectCallback	3-35
PlayerCount	3-31
PlayerSessionKey	3-7
PlayerSkillLevel	3-8, 3-31
pLocalKeyPair	3-39
Port	3-25
pubKey	3-21

R

RulesSet	3-8, 3-31
----------------	-----------

S

ServerIP	3-5
ServerPort	3-5
ServerType	3-25
ServerVersion	3-25
SessionKey	3-24
SpectatorWorldID	3-5, 3-6
Status	3-39

T

TotalActivePlayers	3-24
TrustLevel	3-3

W

WorldID	3-8, 3-12, 3-14, 3-18, 3-33
WorldStatus	3-31

This page intentionally left blank.