



Gatekeeper API Functions and Structures

This chapter describes the API functions and structures that an external application must use to exchange messages with the Cisco IOS Gatekeeper, and contains the following sections:

- [Gatekeeper API Functions, page 5-1](#)
- [API Structures, page 5-11](#)

The external application links with the object code, which contains the API functions. The header file contains API prototypes and type definitions.

Gatekeeper API Functions

This section describes the functions provided with the API. These functions should be used by the external application to gather information from and provide information to the Cisco IOS Gatekeeper. The functions described in this section are:

- [GkapiSetupClient, page 5-2](#)
- [GkapiSetupServer, page 5-2](#)
- [GkapiClientConnected, page 5-3](#)
- [GkapiAcceptConnection, page 5-4](#)
- [GkapiGetVersion, page 5-4](#)
- [CloseGateKeeperConnection, page 5-5](#)
- [GetReadMsgBuffer, page 5-5](#)
- [ReadMsgBuffer, page 5-5](#)
- [FreeReadMsgBuffer, page 5-7](#)
- [WriteResponseMsg, page 5-7](#)
- [WriteRegisterMessage, page 5-8](#)
- [WriteUnregisterMessage, page 5-9](#)
- [GkapiSetupReport, page 5-10](#)
- [GkapiQueryReport, page 5-11](#)

GkapiSetupClient

This function sets up the socket for the application to communicate as a client with the Cisco IOS Gatekeeper. In this situation, the application is the client and the Gatekeeper is the server, which means the application must initiate the communication with the Cisco IOS Gatekeeper.

Input

The input to this function is:

- A pointer to the [GKAPI SOCK INFO](#) structure. The application must set up the TCP Port and IP Address fields and must preserve this structure for the duration of the connection.
- A pointer to the [STATUS_TYPE](#) enumeration. Possible values for STATUS_TYPE are:
 - PROCESSING_SUCCESSFUL—Successful connection to the Cisco IOS Gatekeeper.
 - CONNECT_IN_PROGRESS—Connection is pending.
 - TCP_HANDLE_ERROR—Error was encountered in handle creation.
 - TCP_CONNECT_ERROR—Error was encountered in connecting to the Cisco IOS Gatekeeper.
 - TCP_NONBLOCK_ERROR—Error was encountered when setting up the socket for nonblocking I/O
- A boolean value that allows the application to specify if the socket I/O should be nonblocking or blocking. If the application specifies blocking, the Gatekeeper API calls to setup the connection and read a message that does not return until the action is complete.

Return

The return for this function is an integer. If the client socket connection has been set up successfully or is in progress, a connection handle is returned. This connection handle is the socket descriptor that the application uses to wait on a connection completion or read socket event. If an error occurs while setting up the client connection, the value -1 is returned. In this case, the error information is provided in the STATUS_TYPE.

GkapiSetupServer

This function sets up the socket for the application to communicate as a server with the Cisco IOS Gatekeeper. In this situation, the application is the server and the Gatekeeper is the client, which means that the application will accept incoming connections from Cisco IOS Gatekeeper clients.

Input

- A pointer to the [GKAPI SOCK INFO](#) structure. The application must set up the TCP Port and IP Address fields and must preserve this structure for the duration of the connection.
- A pointer to the [STATUS_TYPE](#) enumeration. Possible values for STATUS_TYPE are:
 - PROCESSING_SUCCESSFUL—Successful connection to the Cisco IOS Gatekeeper.
 - TCP_HANDLE_ERROR—Error was encountered in handle creation.
 - TCP_ADDRESS_ALREADY_IN_USE—Specified local IP address is already in use.

- TCP_ADDRESS_NOT_AVAIL—Specified local IP address is not available on the local machine.
- TCP_BIND_ERROR—Error was encountered in setting up the server socket.
- TCP_LISTEN_ERROR—Error was encountered in setting up the server socket.
- TCP_NONBLOCK_ERROR—Error was encountered when setting up the socket for nonblocking I/O.
- A boolean value that allows the application to specify if the socket I/O should be nonblocking or blocking. If the application specifies blocking, the Gatekeeper API calls to setup the connection and read a message that does not return until the action is complete.

Return

The return for this function is an integer. If the client socket connection has been set up successfully or is in progress, a connection handle is returned. This connection handle is the socket descriptor that the application uses to wait on a connection completion or read socket event. If an error occurs while setting up the client connection, the value -1 is returned. In this case, the error information is provided in the STATUS_TYPE.

GkapiClientConnected

This function must be called by the application to indicate that a select event for a connect complete has occurred.

Input

The input to this function is:

- A pointer to the [GKAPI SOCK_INFO](#) structure.
- A pointer to the STATUS_TYPE enumeration. Possible values for STATUS_TYPE are:
 - PROCESSING_SUCCESSFUL—Successful connection to the Gatekeeper.
 - TCP_CONNECT_ERROR—Error was encountered in connecting to the Gatekeeper.
- An integer that indicates that a connect complete has occurred.

Return

The return for this function is an integer. If the socket connection has been set up successfully or is in progress, a connection handle is returned. This connection handle is the socket descriptor that the application uses to wait on a connection completion or read socket event. If an error occurs while setting up the client connection, the value -1 is returned. In this case, the error information is provided in the STATUS_TYPE.

GkapiAcceptConnection

This function must be called by the application (when it is running in the server mode) to indicate that a select event for an incoming connection has occurred.

Input

The input to this function is

- A pointer to the [GKAPI SOCK_INFO](#) structure.
- A pointer to the STATUS_TYPE enumeration. Possible values for STATUS_TYPE are:
 - PROCESSING_SUCCESSFUL—Successful connection to the Cisco IOS Gatekeeper.
 - TCP_CONNECT_ERROR—Error was encountered in connecting to the Cisco IOS Gatekeeper.
- An integer that indicates that an incoming connection has occurred.
- A pointer to the [GKAPI_TCP_ADDR_INFO](#) structure. The Gatekeeper API provides the IP address and TCP port of the client with which this connection is associated.

Return

The return for this function is an integer. If the socket connection has been set up successfully or is in progress, a connection handle is returned. This connection handle is the socket descriptor that the application uses to wait on a connection completion or read socket event. If an error occurs while setting up the client connection, the value -1 is returned. In this case, the error information is provided in the STATUS_TYPE.

GkapiGetVersion

Applications can use this function to obtain the GKTMP version used by the API and the gatekeeper. The version number consists of a major number (gk_major) and a minor number (gk_minor). For example, Version 1 is represented as 100.

Input

The input to this function is

- A pointer to the [GKAPI_VERSION_INFO](#) structure.

Return

The return for this function is an integer. The integer indicates one of the following status codes:

- GKAPI_RET_OK—Gatekeeper TMP version is prior to or the same as the gatekeeper API version.
- GKAPI_RET_NOK—Gatekeeper TMP version is later than the gatekeeper API version.
- GKAPI_RET_OK_NOGKDATA—Version used by the gatekeeper is not known. In this case, the gk_major and gk_minor members of GKAPI_VERSION_INFO_T are invalid and set to -1.

CloseGateKeeperConnection

This function closes the TCP connection between the external application and the Cisco IOS Gatekeeper. This function is called under error circumstances and when the external application no longer wants to maintain a relationship with the Cisco IOS Gatekeeper.

Input

The input for this function is a pointer to the [GKAPI SOCK_INFO](#) structure.

Return

There is no return for this function.

GetReadMsgBuffer

This function allocates memory for the size of `GK_READ_MSG` structure. This structure is used to store messages received from the Cisco IOS Gatekeeper. This function contains an enumeration of the messages that can be received (REQUEST messages from the Cisco IOS Gatekeeper for RRQ, ARQ, LRQ, LCF, LRJ, as well as registration and unregistration responses from the Cisco IOS Gatekeeper for ARQ, RRQ, URQ, LRQ, LCF, LRJ messages) and a union of structures for the different messages.



Note

When the external application no longer needs the message buffer, the application must call `FreeReadMsgBuffer` to release the memory back to the system.

Input

There is no input to this function.

Return

The return for this function is a pointer to the [GK_READ_MSG](#) structure. If the memory allocation fails, this pointer will be NULL.

ReadMsgBuffer

This function reads a message from the TCP socket and should be called when the external application has detected a read event on the socket. This function stores the message type into the structure. The parameters received in the message are stored in the structure that corresponds with the message type.



Note

`GetReadMsgBuffer` must be called to allocate an empty buffer before this function can be used. `FreeReadMsgBuffer` must be called after this function has completed, except when the `STATUS_TYPE` returns `INCOMPLETE_MSG_READ`.

After reading a message, this function sets the message type and populates the appropriate structure. For example, if an ARQ message has been received from the Cisco IOS Gatekeeper, the `msgType` parameter is set to `ARQ_REQUEST_MSG` and the `ARQ_REQUEST_MSG` structure is populated.

Because some parameters are optional, these parameters might not be received for a particular message. Structure members that are character pointers are initialized to `NULL`. Integers and enumerations are set to their initialization values. Therefore, the API can assume that if a structure member has a pointer set to `NULL` or to its initialization value, that particular parameter has not been received.

The following initialization values indicate that the parameter was not received from the Cisco IOS Gatekeeper:

- `canMapAlias`—`INITIALIZE_CAN_MAP_ALIAS_VALUE`
- `bandWidthPresent`—`TRUE` (indicating that `bandWidth` has been received and filled in) or `FALSE` (indicating that `bandWidth` has not been received)
- `answerCall`—`INITIALIZE_ANSWER_CALL_VALUE`
- `REDIRECT_REASON_TYPE`—`REDIRECT_REASON_INFO_NOT_RCVD`
- `ENDPOINT_TYPE`—`ENDPOINT_INFO_NOT_RCVD`

Input

The input for this function is:

- A pointer to the `GKAPI SOCK_INFO` structure.
- A pointer to the `GK_READ_MSG` structure that was allocated by the `GetReadMsgBuffer` function. The `GK_READ_MSG` structure contains an enumeration of the message types expected from the Cisco IOS Gatekeeper and a union of structures for various messages expected from the Cisco IOS Gatekeeper.

Return

The return for this function is the `STATUS_TYPE`. Possible values for `STATUS_TYPE` are:

- `PROCESSING_SUCCESSFUL`—No errors were encountered.
- `TCP_READ_ERROR`—A TCP read error was encountered. The application should call `CloseGateKeeperConnection` to close the connection to the Cisco IOS Gatekeeper.
- `MEM_ALLOC_FAIL`—Memory allocation failed. This function, dynamically allocates memory for fields within the `GK_READ_MSG` structure.
- `MSG_READ_ERROR`—The message read was not understood by the API function. The application should call `CloseGateKeeperConnection` to close the connection to the Cisco IOS Gatekeeper.
- `INCOMPLETE_MSG_READ`—The message was not completely read from the TCP connection because of network conditions. The application should call the function again in order to continue reading the data. In this situation, `FreeMsgBuffer` should not be called. After all the data has been read, the `STATUS_TYPE` is set to one of the other possible values, and after processing the message type the `FreeMsgBuffer` can be called.
- `TCP_CONNECTION_CLOSED`—The connection to the Cisco IOS Gatekeeper has been closed. The application must call `CloseGateKeeperConnection` to free resources such as `gkHandle` in the `GKAPI SOCK_INFO` structure.
- `NULL_POINTER_PASSED`—The pointer to the `GK_READ_MSG` is null.

FreeReadMsgBuffer

This function frees memory that was allocated by the call to `GetReadMsgBuffer` and `ReadMsgBuffer`. This function **must** be called after processing the information returned by `ReadMsgBuffer`.

Input

The input for this function is a pointer to the `GK_READ_MSG` structure.

Return

There is no return for this function.

WriteResponseMsg

This function writes a response message to the Cisco IOS Gatekeeper. This structure contains `RESPONSE_MSG_TYPE`, which is an enumeration of the response messages that can be sent to the Cisco IOS Gatekeeper.

The calling function must set the message type and populate the appropriate structure within the union. For example, if a response RCF needs to be sent to the Cisco IOS Gatekeeper, the application should set the `msgType` to `RCF_RESPONSE_MSG` and populate the `RCF_RESPONSE_MSG` structure.

The following rules apply to responses sent by the external application to the Cisco IOS Gatekeeper:

- Transport-addresses must be preceded with “I:”, followed by the address.
- Alias-addresses must be preceded with either “H:”, “E:”, or “M:” followed by the alias address.
- Values in a “sequence of values” must be separated by a space.
- `HEADER_INFO` must include the “from”, “to” and “transactionID” fields. The notification field is not used with the `WriteResponseMsg` function.

Only changed or new fields should be populated and sent to the Cisco IOS Gatekeeper. Parameters that are not to be sent to the Cisco IOS Gatekeeper must either be set to their initialization value or to NULL (for pointers). The API assumes that if a structure member is set to its initialization value or has a pointer set to NULL, that parameter should not be sent to the Cisco IOS Gatekeeper.

The following initialization values indicate that the parameter should not be sent to the Cisco IOS Gatekeeper:

- `bandWidthPresent`—TRUE (indicating that the `bandWidth` should be sent) or FALSE (indicating that the `bandWidth` should not be sent)
- `REDIRECT_REASON_TYPE`—`REDIRECT_REASON_INFO_NOT_RCVD`
- `ENDPOINT_TYPE`—`ENDPOINT_INFO_NOT_RCVD`



Note

If the application requires additional time before responding to a message from the Cisco IOS Gatekeeper, the application can send a “delay” message by setting `msgType` to `RIP_RESPONSE_MSG`. The delay value (1 through 65536) must be specified and the `transactionID` must be the same as the one received from the Cisco IOS Gatekeeper.

Input

The input for this function is:

- A pointer to the [GKAPI SOCK_INFO](#) structure.
- A pointer to the [IRR_REQUEST_MSG](#) structure, which contains an enumeration of message types for which a response might be sent to the Cisco IOS Gatekeeper. The input also contains a union of structures for each message response.

Return

The return for this function is the [STATUS_TYPE](#). Possible values for STATUS_TYPE are:

- [PROCESSING_SUCCESSFUL](#)—No errors were encountered.
- [CONNECT_IN_PROGRESS](#)—Connection is pending. The application should retry this API call after some time has passed.
- [TCP_WRITE_ERROR](#)—A TCP write error was encountered. The application should call `CloseGateKeeperConnection` to close the connection to the Cisco IOS Gatekeeper.
- [MEM_ALLOC_FAIL](#)—Memory allocation failed.
- [TCP_CONNECTION_CLOSED](#)—The connection to the Cisco IOS Gatekeeper has been closed. The application must call `CloseGateKeeperConnection` to free resources such as `gkHandle` in the [GKAPI SOCK_INFO](#) structure.
- [INVALID_MSG_SPECIFIED](#)—The message type is not within the [RESPONSE_MSG_TYPE](#) range.
- [INVALID_ENDPOINT_SPECIFIED](#)—The endpoint does not match one of the possible values for [ENDPOINT_TYPE](#).
- [INVALID_REDIRECT_REASON_SPECIFIED](#)—The redirect reason does not match one of the possible values for [REDIRECT_REASON_TYPE](#).
- [INVALID_REJECT_REASON_SPECIFIED](#)—The rejection reason does not match one of the possible values for [REJECT_REASON_TYPE](#).
- [INVALID_DELAY_SPECIFIED](#)—The delay is not within the valid range.
- [HEADER_INFO_INCOMPLETE](#)—One of the fields in the header (To, From, TransactionID) is incomplete.
- [NULL_POINTER_PASSED](#)—The pointer to [GK_WRITE_MSG](#) is null.

WriteRegisterMessage

This function sends a registration message to the Cisco IOS Gatekeeper and allows triggers to be dynamically registered with the Cisco IOS Gatekeeper. This structure, [REGISTER_MSG_TYPE](#), contains an enumeration of messages that can be registered with the Cisco IOS Gatekeeper.

The [REGISTER_REQUEST_HEADER](#) structure must include the “from,” “to,” “priority,” and “notification-only” fields.

Input

The input for this function is:

- A pointer to the [GKAPI_SOCK_INFO](#) structure.
- A pointer to the [GK_REGISTER_MSG](#) structure, which contains a union of the structures for the various registration messages that can be sent to the Cisco IOS Gatekeeper. Each structure contains a header, [REGISTER_REQUEST_HEADER](#), that must be filled in by the application. The `msgType` field must be filled in to indicate which registration message should be sent to the Cisco IOS Gatekeeper.

Return

The return for this function is the [STATUS_TYPE](#). Possible values for `STATUS_TYPE` are:

- `PROCESSING_SUCCESSFUL`—No errors were encountered.
- `CONNECT_IN_PROGRESS`—Connection is pending. The application should retry this API call after some time has passed.
- `TCP_WRITE_ERROR`—A TCP write error was encountered. The application should call `CloseGateKeeperConnection` to close the connection to the Cisco IOS Gatekeeper.
- `MEM_ALLOC_FAIL`—Memory allocation failed.
- `TCP_CONNECTION_CLOSED`—The connection to the Cisco IOS Gatekeeper has been closed. The application must call `CloseGateKeeperConnection` to free resources such as `gkHandle` in the [GKAPI_SOCK_INFO](#) structure.
- `INVALID_MSG_SPECIFIED`—The message type is not within the `RESPONSE_MSG_TYPE` range.
- `INVALID_ENDPOINT_SPECIFIED`—The endpoint does not match one of the possible values for `ENDPOINT_TYPE`.
- `INVALID_REDIRECT_REASON_SPECIFIED`—The redirect reason does not match one of the possible values for `REDIRECT_REASON_TYPE`.
- `HEADER_INFO_INCOMPLETE`—One of the fields in the header (`To`, `From`, `TransactionID`) is incomplete.
- `NULL_POINTER_PASSED`—The pointer to the [GK_REGISTER_MSG](#) is null.

The following initialization values indicate that the parameter should not be sent to the Cisco IOS Gatekeeper for Registration, and the external application is not interested in these parameters:

- `REDIRECT_REASON_TYPE - REDIRECT_REASON_INFO_NOT_RCVD`
- `ENDPOINT_TYPE - ENDPOINT_INFO_NOT_RCVD`

WriteUnregisterMessage

This function sends an unregister message to the Cisco IOS Gatekeeper when the application no longer wants to receive a particular message. This structure contains [REGISTER_MSG_TYPE](#), which is an enumeration of messages that can be unregistered with the Cisco IOS Gatekeeper.

Input

The input for this function is:

- A pointer to the [GKAPI_SOCK_INFO](#) structure.

- A pointer to the [GK_UNREGISTER_MSG](#) structure, which contains the To, From, and Priority fields that must be filled in by the application. The msgType must be filled in to indicate which message needs to be unregistered.

Return

The return for this function is the [STATUS_TYPE](#). Possible values for STATUS_TYPE are:

- [PROCESSING_SUCCESSFUL](#)—No errors were encountered.
- [CONNECT_IN_PROGRESS](#)—Connection is pending. The application should retry this API call after some time has passed.
- [TCP_WRITE_ERROR](#)—A TCP write error was encountered. The application should call `CloseGateKeeperConnection` to close the connection to the Cisco IOS Gatekeeper.
- [MEM_ALLOC_FAIL](#)—Memory allocation failed.
- [TCP_CONNECTION_CLOSED](#)—The connection to the Cisco IOS Gatekeeper has been closed. The application must call `CloseGateKeeperConnection` to free resources such as `gkHandle` in the [GKAPI_SOCK_INFO](#) structure.
- [INVALID_MSG_SPECIFIED](#)—The message type is not within the [RESPONSE_MSG_TYPE](#) range.
- [HEADER_INFO_INCOMPLETE](#)—One of the fields in the header (To, From, TransactionID) is incomplete.
- [NULL_POINTER_PASSED](#)—The pointer to the [GK_UNREGISTER_MSG](#) is null.

GkapiSetupReport

This function allows the application to control the type of debug messages that the Gatekeeper API provides and the location of the debug output.

Input

The input for this function is:

- An integer that indicates the type of debugging. If the debugging is set to 0, the Gatekeeper API will not output any debug messages.
- A pointer to the [REPORT_DEST_T](#) enumeration, which indicates the destination for the debug messages.

Return

There is no return for this function.

GkapiQueryReport

This function returns the current debug setting for the Gatekeeper API.

Input

There is no input for this function.

Return

The return for this function is an integer that indicates the type of debugging being performed by the Gatekeeper API.

API Structures

The Gatekeeper API stores all data received from the Cisco IOS Gatekeeper in structures. The structures point to character strings, integers, and often enumerations (which are lists of possible values for a specific field). The structures used by the Gatekeeper API are:

- [GKAPI SOCK_INFO](#)
- [GKAPI_TCP_ADDR_INFO](#)
- [GKAPI_VERSION_INFO](#)
- [GK_REGISTER_MSG](#)
- [GK_UNREGISTER_MSG](#)
- [REG_UNREG_RESP_MSG](#)
- [REGISTER_REQUEST_HEADER](#)
- [REGISTER_RESPONSE_HEADER](#)
- [ARQ_REGISTER_MSG](#)
- [RRQ_REGISTER_MSG](#)
- [URQ_REGISTER_MSG](#)
- [LRQ_REGISTER_MSG](#)
- [LCF_REGISTER_MSG](#)
- [LRJ_REGISTER_MSG](#)
- [RAI_REGISTER_MSG](#)
- [DRQ_REGISTER_MSG](#)
- [BRQ_REGISTER_MSG](#)
- [GK_READ_MSG](#)
- [HEADER_INFO](#)
- [ARQ_REQUEST_MSG](#)
- [RRQ_REQUEST_MSG](#)
- [URQ_REQUEST_MSG](#)

- LRQ_REQUEST_MSG
- LCF_REQUEST_MSG
- LRJ_REQUEST_MSG
- RAI_REQUEST_MSG
- DRQ_REQUEST_MSG
- BRQ_REQUEST_MSG
- IRR_REQUEST_MSG
- ARQ_RESPONSE_MSG
- ACF_RESPONSE_MSG
- ARJ_RESPONSE_MSG
- RRQ_RESPONSE_MSG
- RCF_RESPONSE_MSG
- RRJ_RESPONSE_MSG
- LRQ_RESPONSE_MSG
- LCF_RESPONSE_MSG
- LRJ_RESPONSE_MSG
- BRQ_RESPONSE_MSG
- BCF_RESPONSE_MSG
- BRJ_RESPONSE_MSG
- ALV_RESPONSE_MSG
- CRYPTO_EP_PWD_HASH
- CRYPTO_EP_PWD_ENCR
- CRYPTO_EP_CERT
- CLEAR_TOKEN
- ALTERNATE_GK
- ALTERNATE_ENDPOINT
- ALTERNATE_TRANSPORT_ADDR_TYPE
- RIP_RESPONSE_MSG
- UNSUPPORTED_MSG

GKAPI SOCK_INFO

The GKAPI SOCK_INFO structure is used by several API functions to identify the connection to the Cisco IOS Gatekeeper. This structure contains the fields shown in [Table 5-1](#):

Table 5-1 GKAPI SOCK_INFO

Field	Field Type	Description
TCPPort	Integer	The TCP port of the Cisco IOS Gatekeeper that is establishing the incoming connection to the application.
IPAddress	Character string	The IP address of the Cisco IOS Gatekeeper that is establishing the incoming connection to the application.
gkHandle	Integer	Handle to the Cisco IOS Gatekeeper function.
serverHandle	Integer	Handle to the server function.

TCPPort and IPAddress are provided by the calling function. The API writes the handle into gkHandle and serverHandle when the connection is established. If an error is encountered in the handle creation or in the connection, the gkHandle will be set to -1. The external application is responsible for storing the handle and using it to read, write, and close the connection.

GKAPI TCP_ADDR_INFO

The GKAPI TCP_ADDR_INFO structure is used to store the TCP Port and IP address. This structure contains the fields shown in [Table 5-2](#):

Table 5-2 GKAPI TCP_ADDR_INFO

Field	Field Type	Description
TCPPort	Integer	The TCP port that the Cisco IOS Gatekeeper uses for handling GKTMP messages. For GkapiSetupServer, this is the TCP port that the application uses for interacting with the Gatekeeper.
IPAddress	Unsigned long	For GkapiSetupClient, this is the IP address that the Cisco IOS Gatekeeper uses for handling GKTMP messages. For GkapiSetupServer, this is the IP address that the application uses for interacting with the Gatekeeper.

GKAPI_VERSION_INFO

The GKAPI_VERSION_INFO structure is used to store the major and minor version numbers of the gatekeeper TMP and API. This structure contains the fields shown in [Table 5-3](#):

Table 5-3 GKAPI_VERSION_INFO

Field	Field Type	Description
gkapi_major	Integer	The major number identifying the version of the API.
gkapi_minor	Integer	The minor number identifying the version of the API.
gktmp_major	Integer	The major number identifying the version of the TMP.
gktmp_minor	Integer	The minor number identifying the version of the TMP.
GKAPI_MAX_VER_STR_LEN	Character string	The build date and target operating system of the protocol.
gkapi_release_num	Integer	The release number of the gatekeeper API.

GK_REGISTER_MSG

The GK_REGISTER_MSG structure is used to send registration messages to the Cisco IOS Gatekeeper. This structure contains the fields shown in [Table 5-4](#):

Table 5-4 GK_REGISTER_MSG

Field	Field Type	Description
msgType	Enumeration	See REGISTER_MSG_TYPE .
rrqRegMsg	Structure	See RRQ_REGISTER_MSG .
urqRegMsg	Structure	See URQ_REGISTER_MSG .
arqRegMsg	Structure	See ARQ_REGISTER_MSG .
lrqRegMsg	Structure	See LRQ_REGISTER_MSG .
lcfRegMsg	Structure	See LCF_REGISTER_MSG .
lrjRegMsg	Structure	See LRJ_REGISTER_MSG .

GK_UNREGISTER_MSG

The GK_UNREGISTER_MSG structure is used to send unregistration messages to the Cisco IOS Gatekeeper. This structure contains the fields shown in [Table 5-5](#):

Table 5-5 GK_UNREGISTER_MSG

Field	Field Type	Description
unregisterMsg	Enumeration	See REGISTER_MSG_TYPE .
versionId	Integer	Identifier of the version of GKTMP being used. For the initial release, the only possible value is 1.

Table 5-5 GK_UNREGISTER_MSG

Field	Field Type	Description
from	Character string	Originator of the message. For requests from the Cisco IOS Gatekeeper, this field contains the gatekeeper ID. For responses from the external application, this field contains the server ID. The limit of this field is MAX_ENDPOINT_LENGTH + 1.
to	Character string	Receiver of the message. For requests from the Cisco IOS Gatekeeper, this field contains the server ID. For responses from the external application, this field contains the ID of the gatekeeper that initiated the request. The limit of this field is MAX_ENDPOINT_LENGTH + 1.
priority	Integer	Priority of the filter. Possible values are 1 through 20. 1 is the highest priority.

REG_UNREG_RESP_MSG

The REG_UNREG_RESP_MSG structure is used to process registration and unregistration responses from the Cisco IOS Gatekeeper. This structure contains the field shown in [Table 5-6](#):

Table 5-6 REG_UNREG_RESP_MSG

Field	Field Type	Description
regHeader	Structure	See REGISTER_RESPONSE_HEADER .

REGISTER_REQUEST_HEADER

The REGISTER_REQUEST_HEADER structure is used when a registration request is to be sent to Cisco IOS Gatekeeper. This structure contains the fields shown in [Table 5-7](#):

Table 5-7 REGISTER_REQUEST_HEADER

Field	Field Type	Description
versionId	Integer	Identifier of the version of GKTMP being used. For the initial release, the only possible value is 1.
from	Character string	Originator of the message, which for registration requests is the server ID. The limit of this field is MAX_ENDPOINT_LENGTH+1.
to	Character string	Receiver of the message, which for registration requests is the gatekeeper ID. The limit of this field is MAX_ENDPOINT_LENGTH+1.
priority	Integer	Priority of the filter. Possible values are 1 through 20. 1 is the highest priority.
notificationOnly	Boolean	Whether the registration request is for notifications only. If this field is set to True, messages that match the specified trigger parameters are sent on a notification-only basis.

REGISTER_RESPONSE_HEADER

The REGISTER_RESPONSE_HEADER structure is used when a registration or unregistration response is received from the Cisco IOS Gatekeeper. The registration or unregistration response is received after the application sends a registration or unregistration request to the Cisco IOS Gatekeeper. This structure contains the fields shown in [Table 5-8](#):

Table 5-8 REGISTER_RESPONSE_HEADER

Field	Field Type	Description
version-id	Integer	Identifier of the version of GKTMP being used. For the initial release, the only possible value is 1.
from	Character string	Originator of the message, which for registration responses is the gatekeeper ID. The limit of this field is MAX_ENDPOINT_LENGTH+1.
to	Character string	Receiver of the message, which for registration responses is the server ID. The limit of this field is MAX_ENDPOINT_LENGTH+1.
priority	Integer	Priority of the filter. Possible values are 1 through 20. 1 is the highest priority.
regStatus	Enumeration	See REG_STATUS_TYPE .

ARQ_REGISTER_MSG

The ARQ_REGISTER_MSG structure is used to send registrations for ARQ requests to the Cisco IOS Gatekeeper.

This structure contains the fields shown in [Table 5-9](#):

Table 5-9 ARQ_REGISTER_MSG

Field	Field Type	Description
headerInfo	Structure	See REGISTER_REQUEST_HEADER .
destinationInfo	Character string	Sequence of alias addresses for the destination endpoint. The limit of this field is MAX_NUM_ARQ_DEST_INFO.
redirectReason	Enumeration	Taken from the Q.931 Setup Redirecting Number IE. See REDIRECT_REASON_TYPE . The limit of this field is MAX_NUM_ARQ_REDIRECT_REASON.

RRQ_REGISTER_MSG

The RRQ_REGISTER_MSG structure is used to send registrations for RRQ requests to the Cisco IOS Gatekeeper. This structure contains the fields shown in [Table 5-10](#):

Table 5-10 RRQ_REGISTER_MSG

Field	Field Type	Description
headerInfo	Structure	See REGISTER_REQUEST_HEADER .
terminalType	Enumeration	Type of endpoint being registered. See ENDPOINT_TYPE . The limit of this field is MAX_NUM_ENDPOINT_TYPES.
supportedPrefix	Character string	Prefix associated with the supported protocol. The limit of this field is MAX_NUM_SUPPORTED_PREFIX.

URQ_REGISTER_MSG

The URQ_REGISTER_MSG structure is used to send registrations for URQ requests to the Cisco IOS Gatekeeper. This structure contains the fields shown in [Table 5-11](#):

Table 5-11 URQ_REGISTER_MSG

Field	Field Type	Description
headerInfo	Structure	See REGISTER_REQUEST_HEADER .
terminalType	Enumeration	Type of endpoint being unregistered. See ENDPOINT_TYPE . The limit of this field is MAX_NUM_ENDPOINT_TYPES.
supportedPrefix	Character string	Prefix associated with the supported protocol. The limit of this field is MAX_NUM_SUPPORTED_PREFIX.

LRQ_REGISTER_MSG

The LRQ_REGISTER_MSG structure is used to send registrations for LRQ requests to the Cisco IOS Gatekeeper.

This structure contains the fields shown in [Table 5-12](#):

Table 5-12 LRQ_REGISTER_MSG

Field	Field Type	Description
headerInfo	Structure	See REGISTER_REQUEST_HEADER .
destinationInfo	Character string	Sequence of alias addresses for the destination endpoint. The limit of this field is MAX_NUM_LRQ_DEST_INFO.
redirectReason	Enumeration	Taken from the Q.931 Setup Redirecting Number IE. See REDIRECT_REASON_TYPE . The limit of this field is MAX_NUM_LRQ_REDIRECT_REASON.

LCF_REGISTER_MSG

The LCF_REGISTER_MSG structure is used to send registrations for LCF requests to the Cisco IOS Gatekeeper. This structure contains the fields shown in [Table 5-13](#):

Table 5-13 LCF_REGISTER_MSG

Field	Field Type	Description
headerInfo	Structure	See REGISTER_REQUEST_HEADER .
destinationInfo	Character string	Sequence of alias addresses for the destination endpoint. The limit of this field is MAX_NUM_LCF_DEST_INFO.
rmotExtensionAddr	Character String	Alias address of a called endpoint, present in cases where this information is required to traverse multiple gateways. The limit of this field is MAX_NUM_LCF_RMOT_EXTENSION_ADDR.

LRJ_REGISTER_MSG

The LRJ_REGISTER_MSG structure is used to send registrations for LRJ requests to the Cisco IOS Gatekeeper. This structure contains the fields shown in [Table 5-14](#):

Table 5-14 LRJ_REGISTER_MSG

Field	Field Type	Description
headerInfo	Structure	See REGISTER_REQUEST_HEADER .
destinationInfo	Character string	Sequence of alias addresses for the destination endpoint. The limit of this field is MAX_NUM_LRJ_DEST_INFO.

RAI_REGISTER_MSG

The RAI_REGISTER_MSG structure is used to send registrations for RAI requests to the Cisco IOS Gatekeeper.

This structure contains the fields shown in [Table 5-15](#):

Table 5-15 RAI_REGISTER_MSG

Field	Field Type	Description
headerInfo	Structure	See REGISTER_REQUEST_HEADER .
terminalType	Enumeration	Type of endpoint. See ENDPOINT_TYPE . The limit of this field is MAX_NUM_ENDPOINT_TYPES.
supportedPrefix	Character string	Prefix associated with the supported protocol. The limit of this field is MAX_NUM_SUPPORTED_PREFIX.

DRQ_REGISTER_MSG

The DRQ_REGISTER_MSG structure is used to send registrations for DRQ requests to the Cisco IOS Gatekeeper. This structure contains the fields shown in [Table 5-16](#):

Table 5-16 RAI_REGISTER_MSG

Field	Field Type	Description
headerInfo	Structure	See REGISTER_REQUEST_HEADER .
destinationInfo	Character string	Sequence of alias addresses for the destination endpoint. The limit of this field is MAX_NUM_ARQ_DEST_INFO.
redirectReason	Enumeration	Taken from the Q.931 Setup Redirecting Number IE. See REDIRECT_REASON_TYPE . The limit of this field is MAX_NUM_LRQ_REDIRECT_REASON.

BRQ_REGISTER_MSG

The BRQ_REGISTER_MSG structure is used to send registrations for BRQ requests to the Cisco IOS Gatekeeper. This structure contains the fields shown in [Table 5-17](#):

Table 5-17 BRQ_REGISTER_MSG

Field	Field Type	Description
headerInfo	Structure	See REGISTER_REQUEST_HEADER .
destinationInfo	Character string	Sequence of alias addresses for the destination endpoint. The limit of this field is MAX_NUM_ARQ_DEST_INFO.
redirectReason	Enumeration	Taken from the Q.931 Setup Redirecting Number IE. See REDIRECT_REASON_TYPE . The limit of this field is MAX_NUM_ARQ_REDIRECT_REASON.

GK_READ_MSG

The GK_READ_MSG structure is used process REQUEST messages from the Cisco IOS Gatekeeper for the supported RAS messages, as well as registration and unregistration responses from the Cisco IOS Gatekeeper for the supported RAS messages. This structure contains the fields shown in [Table 5-18](#):

Table 5-18 GK_READ_MSG

Field	Field Type	Description
msgType	Enumeration	See REQUEST_MSG_TYPE .
rrqReqMsg	Structure	See RRQ_REQUEST_MSG .
urqReqMsg	Structure	See URQ_REQUEST_MSG .
arqReqMsg	Structure	See ARQ_REQUEST_MSG .

Table 5-18 GK_READ_MSG

Field	Field Type	Description
lrqReqMsg	Structure	See LRQ_REQUEST_MSG .
lcfReqMsg	Structure	See LCF_REQUEST_MSG .
lrjReqMsg	Structure	See LRJ_REQUEST_MSG .
raireqMsg	Structure	See RAI_REQUEST_MSG .
drqreqMsg	Structure	See DRQ_REQUEST_MSG .
brqreqMsg	Structure	See BRQ_REQUEST_MSG .
unsupportedMsg	Structure	See UNSUPPORTED_MSG .
regUnregRespMsg	Structure	See REG_UNREG_RESP_MSG .

If the message received from the Cisco IOS Gatekeeper is a RAS message that is not supported by the API function, the msgType is set to MSG_NOT_SUPPORTED. If a response is required, an appropriate response is constructed by the API function and sent to the Cisco IOS Gatekeeper. The header information in the [UNSUPPORTED_MSG](#) structure is filled in by the API function. This situation could occur if the Cisco IOS Gatekeeper has been upgraded to support new messages but the API function has not been correspondingly upgraded.

If the message received from the Cisco IOS Gatekeeper, is not recognized by the API function, the msgType is set to UNKNOWN_MSG and the STATUS_TYPE is set to MSG_READ_ERROR. In this case, the external application should close the connection to the Cisco IOS Gatekeeper by calling the CloseGateKeeperConnection function.

HEADER_INFO

The HEADER_INFO structure is used to process header information sent from the Cisco IOS Gatekeeper or information that is sent by the application to the Cisco IOS Gatekeeper.

This structure contains the fields shown in [Table 5-19](#):

Table 5-19 HEADER_INFO

Field	Field Type	Description
versionId	Integer	Identifier of the version of GKTMP being used. For the initial release, the only possible value is 1.
from	Character string	Originator of the message. For requests from the Cisco IOS Gatekeeper, this field contains the gatekeeper ID. For responses from the external application, this field contains the server ID. The limit of this field is MAX_ENDPOINT_LENGTH + 1.
to	Character string	Receiver of the message. For requests from the Cisco IOS Gatekeeper, this field contains the server ID. For responses from the external application, this field contains the ID of the gatekeeper that initiated the request. The limit of this field is MAX_ENDPOINT_LENGTH+1.

Table 5-19 HEADER_INFO

Field	Field Type	Description
transactionID	Character string	Identifier of the transaction. If this field is present in the request from the Cisco IOS Gatekeeper, it must be echoed in the response from the external application. The limit of this field is MAX_TRANSACTION_ID_LENGTH + 1.
notification	Boolean	Whether the message is for notification purposes only. This field is used only in REQUEST messages that are received from the Cisco IOS Gatekeeper.

ARQ_REQUEST_MSG

The ARQ_REQUEST_MSG structure is used to process ARQ requests from the Cisco IOS Gatekeeper. This structure contains the fields shown in [Table 5-20](#):

Table 5-20 ARQ_REQUEST_MSG

Field	Field Type	Description
headerInfo	Structure	See HEADER_INFO .
srcInfo	Character string	Sequence of alias addresses for the source endpoint.
srcCallSignalAddress	Character string	Transport address used at the source for call signaling.
destinationInfo	Character string	Sequence of alias addresses for the destination endpoint.
destCallSignalAddress	Character string	Transport address used at the destination for call signaling.
destExtraCallInfo	Character string	External addresses for multiple calls.
bandWidthPresent	Boolean	Whether a specified bandwidth is present in the request.
bandWidth	Unsigned integer	Bandwidth (in 100 kbps) requested for the bi-directional call.
answerCall	Integer	Indicates to the Cisco IOS Gatekeeper that the call is incoming.
callIdentifier	Character string	A unique call identifier (set by the originating endpoint), which can be used to associate RAS signaling with the modified Q.931 signaling used in H.225.0.
conferenceID	Character string	A unique conference identifier.
canMapAlias	Integer	Whether the endpoint can copy information from the resulting ACF into the destinationAddress, destExtraCallInfo, and remoteExtensionAddress fields of the SETUP message.
redirectNumber	Character string	Taken from the Number Digits field of Q.931 Setup Redirecting Number IE.
redirectReason	Enumeration	Taken from the Q.931 Setup Redirecting Number IE. See REDIRECT_REASON_TYPE .
callingOctet3a	Character String	Whether the calling number information can be displayed.

Table 5-20 ARQ_REQUEST_MSG

Field	Field Type	Description
displayIE	Character String	Taken from the Q.931 Setup, display IE.
endPointCallSignalAddress	Character String	Call signaling transport address of the endpoint sending the ARQ.
cryptoToken	Pointer	See ALV_RESPONSE_MSG .
clearToken	Pointer	See CLEAR_TOKEN .

RRQ_REQUEST_MSG

The RRQ_REQUEST_MSG structure is used to process RRQ requests from the Cisco IOS Gatekeeper. This structure contains the fields shown in [Table 5-21](#):

Table 5-21 RRQ_REQUEST_MSG

Field	Field Type	Description
headerInfo	Structure	See HEADER_INFO .
callSignalAddress	Character string	Call signaling transport address for this endpoint.
rasAddress	Character string	Registration and status transport address for this endpoint.
terminalAlias	Character string	List of alias addresses by which other terminals can identify this terminal.
terminalType	Enumeration	Type of endpoint being registered. See ENDPOINT_TYPE .
supportedPrefix	Character string	Prefix associated with the supported protocol.
cryptoToken	Pointer	See ALV_RESPONSE_MSG .
clearToken	Pointer	See CLEAR_TOKEN .
altTranspAddr	Pointer	See ALTERNATE_TRANSPORT_ADDR_TYPE .

URQ_REQUEST_MSG

The URQ_REQUEST_MSG structure is used to process URQ requests from the Cisco IOS Gatekeeper. This structure contains the fields shown in [Table 5-22](#):

Table 5-22 URQ_REQUEST_MSG

Field	Field Type	Description
headerInfo	Structure	See HEADER_INFO .
callSignalAddress	Character string	Call signaling transport address for this endpoint.

LRQ_REQUEST_MSG

The LRQ_REQUEST_MSG structure is used to process LRQ requests from the Cisco IOS Gatekeeper. This structure contains the fields shown in [Table 5-23](#):

Table 5-23 LRQ_REQUEST_MSG

Field	Field Type	Description
headerInfo	Structure	See HEADER_INFO .
srcInfo	Character string	Sequence of alias addresses for the source endpoint.
destinationInfo	Character string	Sequence of alias addresses for the destination endpoint.
redirectNumber	Character string	Taken from the Number Digits field of Q.931 Setup Redirecting Number IE.
redirectReason	Enumeration	Taken from the Q.931 Setup Redirecting Number IE. See REDIRECT_REASON_TYPE .
callingOctet3a	Character String	Whether the calling number information can be displayed.
displayIE	Character String	Taken from the Q.931 Setup, display IE.
callingPartyNum	Character String	Taken from the Q.931.

LCF_REQUEST_MSG

The LCF_REQUEST_MSG structure is used to process LCF requests from the Cisco IOS Gatekeeper. This structure contains the fields shown in [Table 5-24](#):

Table 5-24 LCF_REQUEST_MSG

Field	Field Type	Description
headerInfo	Structure	See HEADER_INFO .
srcInfo	Character string	Sequence of alias addresses for the source endpoint.
destinationInfo	Character string	Sequence of alias addresses for the destination endpoint.
callSignalAddress	Character string	Call signaling transport address for this endpoint.
destExtraCallInfo	Character string	External addresses for multiple calls.
redirectNumber	Character string	Taken from the Number Digits field of Q.931 Setup Redirecting Number IE.
redirectReason	Enumeration	Taken from the Q.931 Setup Redirecting Number IE. See REDIRECT_REASON_TYPE .
callingOctet3a	Character String	Whether the calling number information can be displayed.
callingPartyNum	Character String	Taken from the Q.931.
displayIE	Character String	Taken from the Q.931 Setup, display IE.

Table 5-24 LCF_REQUEST_MSG

Field	Field Type	Description
rasAddress	Character String	Registration and status transport address for this endpoint.
rmotExtensionAddr	Character String	Alias address of a called endpoint, present in cases where this information is required to traverse multiple gateways.
destinationType	Enumeration	Type of destination endpoint. See ENDPOINT_TYPE .
altTranspAddr	Pointer	See ALTERNATE_TRANSPORT_ADDR_TYPE.

LRJ_REQUEST_MSG

The LRJ_REQUEST_MSG structure is used to process LRJ requests from the Cisco IOS Gatekeeper. This structure contains the fields shown in [Table 5-25](#):

Table 5-25 LRJ_REQUEST_MSG

Field	Field Type	Description
headerInfo	Structure	See HEADER_INFO .
srcInfo	Character string	Sequence of alias addresses for the source endpoint.
destinationInfo	Character string	Sequence of alias addresses for the destination endpoint.
redirectNumber	Character string	Taken from the Number Digits field of Q.931 Setup Redirecting Number IE.
redirectReason	Enumeration	Taken from the Q.931 Setup Redirecting Number IE. See REDIRECT_REASON_TYPE .
callingOctet3a	Character String	Whether the calling number information can be displayed.
displayIE	Character String	Taken from the Q.931 Setup, display IE.
callingPartyNum	Character String	Taken from the Q.931.
rejectReason	Enumeration	Reason for the rejection of the request. See LRJ_REJECT_REASON_TYPE .

RAI_REQUEST_MSG

The RAI_REQUEST_MSG structure is used to process RAI requests from the Cisco IOS Gatekeeper. This structure contains the fields shown in [Table 5-26](#):

Table 5-26 RAI_REQUEST_MSG

Field	Field Type	Description
headerInfo	Structure	See REGISTER_REQUEST_HEADER .
callSignalAddress	Character string	Call signaling transport address for this endpoint.
almostOut	Integer	Resource usage indication. The value is TRUE or FALSE.

DRQ_REQUEST_MSG

The DRQ_REQUEST_MSG structure is used to process DRQ requests from the Cisco IOS Gatekeeper. This structure contains the fields shown in [Table 5-27](#):

Table 5-27 DRQ_REQUEST_MSG

Field	Field Type	Description
headerInfo	Structure	See REGISTER_REQUEST_HEADER .
drqReason	Enumeration	Reason received for a DRQ sent by an endpoint. See DRQ_REASON_TYPE .
srcCallSignalAddress	Character string	Transport address used at the source for call signaling.
answeredCall	Integer	Indicates that this party was the original destination. The value is TRUE or FALSE.
callIdentifier	Character string	A unique call identifier (set by the originating endpoint) which can be used to associate RAS signaling with the modified Q.931 signaling used in H225.0.
conferenceID	Character string	A unique identifier.
clearToken	Pointer	See CLEAR_TOKEN .

BRQ_REQUEST_MSG

The BRQ_REQUEST_MSG structure is used to process BRQ requests from the Cisco IOS Gatekeeper. This structure contains the fields shown in [Table 5-28](#):

Table 5-28 BRQ_REQUEST_MSG

Field	Field Type	Description
headerInfo	Structure	See HEADER_INFO .
answerCall	Integer	Indicates to the Cisco IOS Gatekeeper that the call is incoming.

Table 5-28 BRQ_REQUEST_MSG

Field	Field Type	Description
bandWidth	Unsigned integer	Bandwidth (in 100 kbps) requested for the bi-directional call.
callIdentifier	Character string	A unique call identifier (set by the originating endpoint), which can be used to associate RAS signaling with the modified Q.931 signaling used in H.225.0.
conferenceID	Character string	A unique conference identifier.
endPointCallSignal Address	Character string	Call signalling transport address for this endpoint.
cryptoToken	Pointer	See ALV_RESPONSE_MSG .
clearToken	Pointer	See CLEAR_TOKEN .

IRR_REQUEST_MSG

The IRR_REQUEST_MSG structure is used to process IRR requests from the Cisco IOS Gatekeeper. This structure contains the fields shown in [Table 5-29](#).

Table 5-29 IRR_REQUEST_MSG

Field	Field Type	Description
srcCallSignalAddress	Character String	Transport address used at the source for call signaling.
perCallInfo	Character String	Contains information about the call, including the call identifier, conference ID, originator, bandwidth, and call start time.

ALV_REQUEST_MSG

The ALV_REQUEST_MSG structure is used to process ALV requests from the Cisco IOS Gatekeeper. This structure contains the fields shown in [Table 5-30](#):

Table 5-30 ALV_REQUEST_MSG

Field	Field Type	Description
headerInfo	Structure	See HEADER_INFO .

GK_WRITE_MSG

The GK_WRITE_MSG structure is used to process responses from the external application to the Cisco IOS Gatekeeper.

This structure contains the fields shown in [Table 5-31](#):

Table 5-31 GK_WRITE_MSG

Field	Field Type	Description
msgType	Enumeration	See RESPONSE_MSG_TYPE .
arqRespMsg	Structure	See ARQ_RESPONSE_MSG .
acfRespMsg	Structure	See ACF_RESPONSE_MSG .
arjRespMsg	Structure	See ARJ_RESPONSE_MSG .
rrqRespMsg	Structure	See RRQ_RESPONSE_MSG .
rrjRespMsg	Structure	See RRJ_RESPONSE_MSG .
rcfRespMsg	Structure	See RCF_RESPONSE_MSG .
lrqRespMsg	Structure	See LRQ_RESPONSE_MSG .
lcfRespMsg	Structure	See LCF_RESPONSE_MSG .
lrjRespMsg	Structure	See LRJ_RESPONSE_MSG .
ripRespMsg	Structure	See RIP_RESPONSE_MSG .

ARQ_RESPONSE_MSG

The ARQ_RESPONSE_MSG structure is used to process ARQ responses from the external application. This structure contains the fields shown in [Table 5-32](#):

Table 5-32 ARQ_RESPONSE_MSG

Field	Field Type	Description
headerInfo	Structure	See HEADER_INFO .
destinationInfo	Character string	Sequence of alias addresses for the destination endpoint.
destCallSignalAddress	Character string	Transport address used at the destination for call signaling.
destExtraCallInfo	Character string	External addresses for multiple calls.
bandWidthPresent	Boolean	Whether a specified bandwidth is present in the request.
bandWidth	Unsigned integer	Bandwidth (in 100 kbps) requested for the bidirectional call.
redirectNumber	Character string	Taken from the Number Digits field of Q.931 Setup Redirecting Number IE.

Table 5-32 ARQ_RESPONSE_MSG

Field	Field Type	Description
redirectReason	Enumeration	Taken from the Q.931 Setup Redirecting Number IE. See REDIRECT_REASON_TYPE .
displayIE	Character String	Taken from the Q.931 Setup, display IE.

ACF_RESPONSE_MSG

The ACF_RESPONSE_MSG structure is used to process ACF responses from the external application. This structure contains the fields shown in [Table 5-33](#):

Table 5-33 ACF_RESPONSE_MSG

Field	Field Type	Description
headerInfo	Structure	See HEADER_INFO .
destinationInfo	Character string	Sequence of alias addresses for the destination endpoint.
destCallSignalAddress	Character string	Transport address used at the destination for call signaling.
destExtraCallInfo	Character string	External addresses for multiple calls.
rmtExtensionAddr	Character string	Alias address of a called endpoint, present in cases where this information is required to traverse multiple gateways.
bandWidthPresent	Boolean	Whether a specified bandwidth is present in the request.
bandWidth	Unsigned integer	Bandwidth (in 100 kbps) requested for the bidirectional call.
destinationType	Enumeration	Type of destination endpoint. See ENDPOINT_TYPE .
altEndpt	Structure	See ALTERNATE_ENDPOINT .
clearToken	Pointer	See CLEAR_TOKEN .
altTranspAddr	Pointer	See ALTERNATE_TRANSPORT_ADDR_TYPE .
use_transport	Enumeration	See USE_SPECIFIED_TRANSPORT_TYPE_T .

ARJ_RESPONSE_MSG

The ARJ_RESPONSE_MSG structure is used to process ARJ responses from the external application. This structure contains the fields shown in [Table 5-34](#):

Table 5-34 ARJ_RESPONSE_MSG

Field	Field Type	Description
headerInfo	Structure	See HEADER_INFO .
rejectReason	Enumeration	Reason the request was rejected. See ARJ_REJECT_REASON_TYPE .

RRQ_RESPONSE_MSG

The RRQ_RESPONSE_MSG structure is used to process RRQ responses from the external application. This structure contains the fields shown in [Table 5-35](#):

Table 5-35 RRQ_RESPONSE_MSG

Field	Field Type	Description
headerInfo	Structure	See HEADER_INFO .
terminalAlias	Character string	List of alias addresses by which other terminals can identify this terminal.
supportedPrefix	Character string	Prefix associated with the supported protocol.

RCF_RESPONSE_MSG

The RCF_RESPONSE_MSG structure is used to process RCF responses from the external application. This structure contains the fields shown in [Table 5-36](#):

Table 5-36 RCF_RESPONSE_MSG

Field	Field Type	Description
headerInfo	Structure	See HEADER_INFO .
terminalAlias	Character string	List of alias addresses by which other terminals can identify this terminal.
supportedPrefix	Character string	Prefix associated with the supported protocol.
alternateGK	Structure	See ALTERNATE_GK .

RRJ_RESPONSE_MSG

The RRJ_RESPONSE_MSG structure is used to process RRJ responses from the external application. This structure contains the fields shown in [Table 5-37](#):

Table 5-37 RRJ_RESPONSE_MSG

Field	Field Type	Description
headerInfo	Structure	See HEADER_INFO .
rejectReason	Enumeration	Reason the request was rejected. See RRJ_REJECT_REASON_TYPE .

LRQ_RESPONSE_MSG

The LRQ_RESPONSE_MSG structure is used to process LRQ responses from the external application. This structure contains the fields shown in [Table 5-38](#):

Table 5-38 LRQ_RESPONSE_MSG

Field	Field Type	Description
headerInfo	Structure	See HEADER_INFO .
destinationInfo	Character string	Sequence of alias addresses for the destination endpoint.

LCF_RESPONSE_MSG

The LCF_RESPONSE_MSG structure is used to process LCF responses from the external application. This structure contains the fields shown in [Table 5-39](#):

Table 5-39 LCF_RESPONSE_MSG

Field	Field Type	Description
headerInfo	Structure	See HEADER_INFO .
destinationInfo	Character string	Sequence of alias addresses for the destination endpoint.
destExtraCallInfo	Character string	External addresses for multiple calls.
callSignalAddress	Character string	Call signaling transport address for this endpoint.
rasAddress	Character String	Registration and status transport address for this endpoint.
rmotExtensionAddr	Character String	Alias address of a called endpoint, present in cases where this information is required to traverse multiple gateways.
destinationType	Enumeration	Type of destination endpoint. See ENDPOINT_TYPE .
altTranspAddr	Pointer	See ALTERNATE_TRANSPORT_ADDR_TYPE .

LRJ_RESPONSE_MSG

The LRJ_RESPONSE_MSG structure is used to process LRJ responses from the external application. This structure contains the fields shown in [Table 5-40](#):

Table 5-40 LRJ_RESPONSE_MSG

Field	Field Type	Description
headerInfo	Structure	See HEADER_INFO .
rejectReason	Enumeration	Reason the request was rejected. See LRJ_REJECT_REASON_TYPE .

BRQ_RESPONSE_MSG

The BRQ_RESPONSE_MSG structure is used to process BRQ responses from the external application. This structure contains the fields shown in [Table 5-41](#):

Table 5-41 BRQ_RESPONSE_MSG

Field	Field Type	Description
headerInfo	Structure	See HEADER_INFO .
bandWidth	Unsigned integer	Bandwidth (in 100 kbps) requested for the bidirectional call.

BCF_RESPONSE_MSG

The BCF_RESPONSE_MSG structure is used to process BCF responses from the external application. This structure contains the fields shown in [Table 5-42](#):

Table 5-42 BCF_RESPONSE_MSG

Field	Field Type	Description
headerInfo	Structure	See HEADER_INFO .
bandWidth	Unsigned integer	Bandwidth (in 100 kbps) requested for the bidirectional call.

BRJ_RESPONSE_MSG

The BRJ_RESPONSE_MSG structure is used to process BRJ responses from the external application. This structure contains the fields shown in [Table 5-43](#):

Table 5-43 BRJ_RESPONSE_MSG

Field	Field Type	Description
headerInfo	Structure	See HEADER_INFO .
rejectReason	Enumeration	See BRJ_REJECT_REASON_TYPE .

ALV_RESPONSE_MSG

The ALV_RESPONSE_MSG structure is used to process ALV responses from the external application. This structure contains the fields shown in [Table 5-44](#):

Table 5-44 ALV_RESPONSE_MSG

Field	Field Type	Description
headerInfo	Structure	See HEADER_INFO .

CRYPTO_H323_TOKEN

The CRYPTO_H323_TOKEN structure is used to process cryptoTokens. This structure contains the fields shown in [Table 5-45](#):

Table 5-45 CRYPTO_H323_TOKEN

Field	Field Type	Description
token_type	Enumeration	See CRYPTO_H323_TOKEN_TYPE_S .
cryptoEPPwdHash	Structure	See CRYPTO_EP_PWD_HASH .
cryptoEPPwdEncr	Structure	See CRYPTO_EP_PWD_ENCR .
cryptoEPCert	Structure	See CRYPTO_EP_CERT .

CRYPTO_EP_PWD_HASH

The CRYPTO_EP_PWD_HASH structure is used to process cryptoTokens. This structure contains the sections shown in [Table 5-46](#):

Table 5-46 CRYPTO_EP_PWD_HASH

Field	Field Type	Description
alias	Character string	Registration and status transport address for this endpoint.
timestamp	Character string	32-bit integer that represents UTC time.
token	Character string	16 octet IA5String that represents the MD5 hashed encoded PwdCertToken.

CRYPTO_EP_PWD_ENCR

The CRYPTO_EP_PWD_ENCR structure is used to process the encrypted data of a cryptoToken. This structure contains the fields shown in [Table 5-47](#):

Table 5-47 CRYPTO_EP_PWD_ENCR

Field	Field Type	Description
paramS	Character string	Any runtime parameters.
encryptedData	Character string	Encrypted data from the cryptoToken.

CRYPTO_EP_CERT

The CRYPTO_EP_CERT structure is used to process the authentication certificate of a cryptoToken. This structure contains the fields shown in [Table 5-48](#):

Table 5-48 CRYPTO_EP_CERT

Field	Field Type	Description
toBeSigned	Character string	Whether the certificate requires a signature.
signature	Character string	Digital signature assigned to the authentication certificate.

CLEAR_TOKEN

The CLEAR_TOKEN structure is used to process the clear tokens field. This structure contains the fields shown in [Table 5-49](#):

Table 5-49 CLEAR_TOKEN

Field	Field Type	Description
objectIdentifier	Character string	Object identifier.
password	Character string	Secret character string that is used to authenticate a user or H.323 endpoint.
timestamp	Character string	32-bit integer that represents UTC time.
challengeString	Character string	Challenge string used for authentication.
random	Character string	Integer value, for example a monotonically increasing sequence number.
generalID	Character string	Character string that uniquely identifies either the sender or receiver.
nonstd_objectID	Character string	Object identifier that is used to indicate the type and format of the nonstandard data being sent in the clear token.
nonstd_data	Character string	Nonstandard data in the clear tokens field.

ALTERNATE_GK

The ALTERNATE_GK structure is used to process information about an alternate gatekeeper. This structure contains the fields shown in [Table 5-50](#):

Table 5-50 ALTERNATE GK

Field	Field Type	Description
rasAddress	Character string	Registration and status transport address for this endpoint.
gkIdentifier	Character string	Identifier of the gatekeeper.
needToRegister	Boolean	Whether there is a need to register with this gatekeeper.
priority	Integer	Priority of this gatekeeper. Possible values are 1 through 127.

ALTERNATE_ENDPOINT

The ALTERNATE_ENDPOINT structure is used to process information about an alternate H.323 endpoint.

This structure contains the fields shown in [Table 5-51](#):

Table 5-51 ALTERNATE_ENDPOINT

Field	Field Type	Description
callSignalAddress	Character string	Registration and status transport address for this endpoint.
tokenP	Structure	See CLEAR_TOKEN .

ALTERNATE_TRANSPORT_ADDR_TYPE

The ALTERNATE_TRANSPORT_ADDR_TYPE structure is used to convey information about an Annex E transport address of the destination H.323 endpoint. This structure contains the fields shown in [Table 5-52](#):

Table 5-52 ALTERNATE_TRANSPORT_ADDR_TYPE

Field	Field Type	Description
annexE	Character string	Annex E transport address of the destination endpoint.
nextP	Pointer	Pointer to the next node in the linked list.

RIP_RESPONSE_MSG

The RIP_RESPONSE_MSG structure is used to process requests from the external application for additional time. This structure contains the fields shown in [Table 5-53](#):

Table 5-53 RIP_RESPONSE_MSG

Field	Field Type	Description
headerInfo	Structure	See HEADER_INFO .
delay	Integer	Amount of time, in milliseconds (1 through 65,536), that the endpoint should wait before retrying the request.

UNSUPPORTED_MSG

The UNSUPPORTED_MSG structure is used to process requests from the Cisco IOS Gatekeeper that contain a RAS message type that is not supported by the API. This structure contains the field shown in [Table 5-54](#):

Table 5-54 UNSUPPORTED MSG

Field	Field Type	Description
headerInfo	Structure	See HEADER_INFO .

Enumerations

Some of the API structures contain enumerations. An enumeration is simply a list of possible values. This section lists the enumerations used by the structures and includes the following sections:

- [STATUS_TYPE](#), page 5-36
- [REG_STATUS_TYPE](#), page 5-36
- [ENDPOINT_TYPE](#), page 5-37
- [REDIRECT_REASON_TYPE](#), page 5-37
- [DRQ_REASON_TYPE](#), page 5-37
- [LRJ_REJECT_REASON_TYPE](#), page 5-37
- [REQUEST_MSG_TYPE](#), page 5-38
- [RRJ_REJECT_REASON_TYPE](#), page 5-39
- [ARJ_REJECT_REASON_TYPE](#), page 5-39
- [BRJ_REJECT_REASON_TYPE](#), page 5-39
- [RESPONSE_MSG_TYPE](#), page 5-39
- [REGISTER_MSG_TYPE](#), page 5-40
- [REPORT_DEST_T](#), page 5-40
- [CRYPTO_H323_TOKEN_TYPE_S](#), page 5-40
- [USE_SPECIFIED_TRANSPORT_TYPE_T](#), page 5-41

STATUS_TYPE

The STATUS_TYPE enumeration lists the possible return values from calls to read, write, register and unregister functions. The possible values are:

- PROCESSING_SUCCESSFUL
- CONNECT_IN_PROGRESS
- NULL_POINTER_PASSED
- TCP_HANDLE_ERROR
- TCP_CONNECT_ERROR
- TCP_READ_ERROR
- TCP_BIND_ERROR
- TCP_LISTEN_ERROR
- TCP_ADDRESS_ALREADY_IN_USE
- TCP_ADDRESS_NOT_AVAIL
- TCP_NONBLOCK_ERROR
- MEM_ALLOC_FAIL
- MSG_READ_ERROR
- TCP_WRITE_ERROR
- TCP_CONNECTION_CLOSED
- INCOMPLETE_MSG_READ
- INVALID_MSG_SPECIFIED
- INVALID_ENDPOINT_SPECIFIED
- INVALID_REDIRECT_REASON_SPECIFIED
- INVALID_REJECT_REASON_SPECIFIED
- INVALID_DELAY_SPECIFIED
- HEADER_INFO_INCOMPLETE

REG_STATUS_TYPE

The REG_STATUS_TYPE enumeration lists the possible status values for registration and unregistration responses received from the Cisco IOS Gatekeeper. The possible values are:

- SUCCESSFUL
- INVALID_PRIORITY
- INVALID_FILTERS
- INVALID_GKID

ENDPOINT_TYPE

The ENDPOINT_TYPE enumeration lists the possible types of endpoints. The possible values are:

- GATEKEEPER
- TERMINAL
- MCU
- PROXY
- VOICEGATEWAY
- H320GATEWAY
- OTHERGATEWAY
- ENDPOINT_INFO_NOT_RCVD

REDIRECT_REASON_TYPE

The REDIRECT_REASON_TYPE enumeration lists the possible reasons that a call might be redirected. The possible values are:

- REDIRECT_REASON_UNKNOWN = 0
- REDIRECT_REASON_CALL_FWD_BUSY = 1
- REDIRECT_REASON_CALL_FWD_NO_REPLY = 2
- REDIRECT_REASON_CALL_DEFLECTION = 4
- REDIRECT_REASON_CLED_DTE_OUT_OF_ORDER = 9
- REDIRECT_REASON_CALL_FWDING_BY_CLED_DTE = 10
- REDIRECT_REASON_CALL_FWDING_UNCONDL = 15
- REDIRECT_REASON_INFO_NOT_RCVD = 99

DRQ_REASON_TYPE

The DRQ_REASON_TYPE enumeration lists the reasons received for a DRQ sent by an endpoint. The possible values are:

- DRQ_REASON_FORCED_DROP = 1
- DRQ_REASON_NORMAL_DROP = 2
- DRQ_REASON_UNDEF_REASON = 3

LRJ_REJECT_REASON_TYPE

The LRJ_REJECT_REASON_TYPE enumeration lists the possible reasons that an LRQ request might be rejected. The possible values are:

- LRJ_NOT_REGISTERED
- LRJ_INVALID_PERMISSION
- LRJ_REQUEST_DENIED
- LRJ_UNDEFINED_REASON
- LRJ_SECURITY_DENIAL

REQUEST_MSG_TYPE

The REQUEST_MSG_TYPE enumeration lists the possible messages that can be received from the Cisco IOS Gatekeeper. The possible values are:

- UNKNOWN_MSG
- MSG_NOT_SUPPORTED
- RRQ_REQUEST_MSG
- URQ_REQUEST_MSG
- ARQ_REQUEST_MSG
- LRQ_REQUEST_MSG
- LRJ_REQUEST_MSG
- LCF_REQUEST_MSG
- BRQ_REQUEST_MSG
- RAI_REQUEST_MSG
- DRQ_REQUEST_MSG
- RRQ_REGISTER_RESPONSE_MSG
- URQ_REGISTER_RESPONSE_MSG
- ARQ_REGISTER_RESPONSE_MSG
- LRQ_REGISTER_RESPONSE_MSG
- LCF_REGISTER_RESPONSE_MSG
- LRJ_REGISTER_RESPONSE_MSG
- BRQ_REGISTER_RESPONSE_MSG
- RAI_REGISTER_RESPONSE_MSG
- DRQ_REGISTER_RESPONSE_MSG
- RRQ_UNREGISTER_RESPONSE_MSG
- URQ_UNREGISTER_RESPONSE_MSG
- ARQ_UNREGISTER_RESPONSE_MSG
- LRQ_UNREGISTER_RESPONSE_MSG
- LCF_UNREGISTER_RESPONSE_MSG
- LRJ_UNREGISTER_RESPONSE_MSG
- BRQ_UNREGISTER_RESPONSE_MSG
- RAI_UNREGISTER_RESPONSE_MSG
- DRQ_UNREGISTER_RESPONSE_MSG

RRJ_REJECT_REASON_TYPE

The RRJ_REJECT_REASON_TYPE enumeration lists the possible reasons that an RRQ request might be rejected. The possible values are:

- RRJ_UNDEFINED_REASON
- RRJ_SECURITY_DENIAL
- RRJ_RESOURCE_UNAVAIL

ARJ_REJECT_REASON_TYPE

The ARJ_REJECT_REASON_TYPE enumeration lists the possible reasons that an ARQ request might be rejected. The possible values are:

- CALLED_PARTY_NOT_REGISTERED
- INVALID_PERMISSION
- REQUEST_DENIED
- UNDEFINED_REASON
- ARJ_RESOURCE_UNAVAIL
- ARJ_SECURITY_DENIAL

BRJ_REJECT_REASON_TYPE

The BRJ_REJECT_REASON_TYPE enumeration lists the possible reasons that a BRQ request might be rejected. The possible values are:

- BRJ_NOT_BOUND
- BRJ_INVALID_CONF_ID
- BRJ_INVALID_PERMISSION
- BRJ_INSUFFICIENT_RSC
- BRJ_INVALID_REVISION
- BRJ_UNDEFINED_REASON
- BRJ_SECURITY_DENIAL

RESPONSE_MSG_TYPE

The RESPONSE_MSG_TYPE enumeration lists the possible messages that the external application can send to the Cisco IOS Gatekeeper. The possible values are:

- RRQ_RESPONSE_MSG
- RCF_RESPONSE_MSG
- RRJ_RESPONSE_MSG
- ARQ_RESPONSE_MSG
- ACF_RESPONSE_MSG
- ARJ_RESPONSE_MSG
- LRQ_RESPONSE_MSG

- LCF_RESPONSE_MSG
- LRJ_RESPONSE_MSG
- RIP_RESPONSE_MSG
- BRQ_RESPONSE_MSG
- BCF_RESPONSE_MSG
- BRJ_RESPONSE_MSG

REGISTER_MSG_TYPE

The REGISTER_MSG_TYPE enumeration lists the possible registration messages that the external application can send to the Cisco IOS Gatekeeper. The possible values are:

- RRQ_REGISTER_MSG
- URQ_REGISTER_MSG
- ARQ_REGISTER_MSG
- LRQ_REGISTER_MSG
- LCF_REGISTER_MSG
- LRJ_REGISTER_MSG
- BRQ_REGISTER_MSG
- RAI_REGISTER_MSG
- DRQ_REGISTER_MSG

REPORT_DEST_T

The REPORT_DEST_T enumeration lists the possible destinations for the Gatekeeper API debug output. The possible values are:

- REPORT_CONSOLE
- REPORT_SYSLOG

CRYPTO_H323_TOKEN_TYPE_S

The CRYPTO_H323_TOKEN_TYPE_S enumeration lists the possible types of cryptoTokens. The possible values are:

- NO_CRYPTO_TOKEN
- CRYPTO_EP_PWD_HASH
- CRYPTO_EP_PWD_ENCR
- CRYPTO_EP_CERT



Note

In the first release of the GKTMP and API, the CRYPTO_EP_PWD_HASH is the only type of cryptoToken supported.

USE_SPECIFIED_TRANSPORT_TYPE_T

The USE_SPECIFIED_TRANSPORT_TYPE_T enumeration lists the possible transport types that an endpoint can select for H.225 signalling. The possible values are:

- TRANSPORT_NONE
- ANNEX_E
- TCP

Limits

Some of the fields are limited in size. The limits are set using variables in the header file. The limits as set in the default header file are shown in [Table 5-55](#):

Table 5-55 Field Size Limits

Variable	Initial Value
MAX_IP_ADDR_LENGTH	15
MAX_VERSION_ID_LENGTH	4
MAX_ENDPOINT_LENGTH	128
MAX_TRANSACTION_ID_LENGTH	24
MAX_NUM_ENDPOINT_TYPES	7
MAX_NUM_SUPPORTED_PREFIX	10
MAX_NUM_ARQ_DEST_INFO	20
MAX_NUM_ARQ_REDIRECT_REASON	7
MAX_NUM_LRQ_DEST_INFO	20
MAX_NUM_LRQ_REDIRECT_REASON	7
MAX_NUM_LCF_DEST_INFO	20
MAX_NUM_LCF_RMOT_EXTENSION_ADDR	20
MAX_NUM_LRJ_DEST_INFO	20
MAX_CRYPT_TOKEN_FIELDS	5

