FUJ00098220

## RiposteCreateMessageEx

The **RiposteCreateMessageEx** function creates a message.

**unsigned long RiposteCreateMessageEx (**
**unsigned long** *dwGroupId*,
**unsigned char** **szMessage,*
**unsigned long** *dwMessageFlags,*
**unsigned long** *dwTranAction,*
**TRAN_HANDLE** **phTran,*
**unsigned long** **pdwNum,*
**unsigned long** *dwExpiry*
**);**

**Parameters**

*dwGroupId*
The group identifier of the group in which the message will be created. If the host message server is not a correspondence server, specify zero to use the group of the host message server.

*szMessage*
A null-terminated string containing a list of message attributes in attribute grammar format (see Attribute Grammar). This list must not contain any attributes which use reserved attribute names. See Riposte32 Constants for a list of reserved attribute names. To avoid potential conflicts with reserved attribute names, it is recommended that applications encapsulate their attributes within an attribute named **RIPOSTE_DATA_ATTRIBUTE**, which is explicitly not reserved.

*dwMessageFlags*
A bitmask of message creation flags. If no message creation flags are specified, the message will be created as a normal user message (see Remarks below). Message creation flags can be combined using a bitwise OR. The following table lists the message creation flags which are currently defined:

| | |
|---|---|
| **RIPOSTE_SYSTEM_MESSAGE** | The message to be created is a system message and will not include an attribute identifying the user creating the message. This is typically used for messages created by applications at startup time or at other times when there may be no user logged on. |
| **RIPOSTE_TEST_MESSAGE** | The message to be created is a test message. All of the message attributes specified in *szMessage* will be encapsulated in an attribute named **RIPOSTE_TESTDATA_ATTRIBUTE**. Test messages are currently treated as system messages, whether the **RIPOSTE_SYSTEM_MESSAGE** flag is set or not. For future compatibility, it is recommended that **RIPOSTE_SYSTEM_MESSAGE** be set whenever **RIPOSTE_TEST_MESSAGE** is set. |

FUJ00098220

*dwTranAction*

A transaction flag which specifies transaction management. The following table lists possible values for this flag:

| | |
|---|---|
| **RIPOSTE_TRAN_ATOMIC** | A transaction will be created solely for the purpose of creating this single message and then committed. *phTran* must specify a pointer to NULL, and no transaction handle will be returned. |
| **RIPOSTE_TRAN_START** | A new transaction will be started for this message. *phTran* must specify a pointer to NULL and will point to the handle of the new transaction upon return. |
| **RIPOSTE_TRAN_NEXT** | The message will be created within the context of a previously created transaction. *phTran* must specify a pointer to a transaction handle created by RiposteStartTransaction or a previous call to **RiposteCreateMessageEx**. |
| **RIPOSTE_TRAN_END** | The message will be created within the context of a previously created transaction, and this transaction will then be committed. *phTran* must specify a pointer to a transaction handle created by RiposteStartTransaction or a previous call to **RiposteCreateMessageEx**, and will point to NULL upon return. |

*phTran*

A pointer to a transaction handle which was created by RiposteStartTransaction (or a previous call to **RiposteCreateMessageEx**), or a pointer to NULL, depending on the value of *dwTranAction*. Transaction handles are RPC context handles (see below).

*pdwNum*

A pointer to a longword which will receive the message sequence number of the created message. The sequence number will be one greater than the sequence number of the last message created by the host message server for this group. This parameter is optional and may be NULL.

*dwExpiry*

Specifies the message expiry of the message in days. The message expiry determines when the message will expire and become subject to archiving. The message will expire at the end of the day (in UTC time) *dwExpiry* days past the current date (in UTC time). If *dwExpiry* is zero, then the message expiry will be set according to the configuration parameter DefaultMessageExpiry. Note that if *dwExpiry* is less than the configuration parameter MinMessageExpiry, then the message expiry will be set to MinMessageExpiry, and if *dwExpiry* is greater than the configuration parameter MaxMessageExpiry, then the message expiry will be set to MaxMessageExpiry.

FUJ00098220

## Return Values

The function returns 0 if successful, otherwise it returns an error code which may be an RPC error code or a Windows NT error code. Use RiposteErrorString to get a text description of the error. The most common return values are listed in the table below:

| | |
|---|---|
| **MSG_RIPOSTE_OFFLINE** | The message server is offline. New messages cannot be created until the message server is online. Use RiposteGetSystemInfo to check the state of the message server. |
| **MSG_RIPOSTE_NO_USER_LOGGED_ON** | Attempt to create a user message when no user is logged on. |
| **MSG_RIPOSTE_USER_RIGHT_NOT_HELD** | The current user does not possess the **RIPOSTE_CREATE_MESSAGES_USER_RIGHT**. This user right is required in order to create user messages. |
| **MSG_RIPOSTE_USER_OPERATION_NOT_PERMITTED** | Attempt to create a user message on a correspondence server. Only system messages can be created on a correspondence server. |
| **MSG_RIPOSTE_INVALID_TRANSACTION_HANDLE** | The transaction handle pointed to by *phTran* is invalid, or the transaction handle was created in a different group from the group in which the message is being created. |

## Remarks

**RiposteCreateMessageEx** creates a message. Normally, the message created is a user message (*dwMessageFlags* is set to zero), in which case there must be a user currently logged on to the message server, and the user must possess the **RIPOSTE_CREATE_MESSAGES_USER_RIGHT**. Alternatively, the message may be a system message, in which case no user need be logged on, and if there is a user logged on they do not need to possess the **RIPOSTE_CREATE_MESSAGE_USER_RIGHT**. System messages are generally used for messages generated by the application which do not correspond to user activity, e.g. a message written at startup or as the result of background tasks.

The message that is created will be formed by taking the message attributes specified in *szMessage* and adding standard Riposte message attributes, such as the current date and time, the group identifier, node identifier, and message sequence number, the user name of the user creating the message (if it is a user message), and a cyclic redundancy check (CRC). Note that the message is uniquely identified by the combination of group identifier, node identifier and message sequence number, where the group identifier is specified by *dwGroupId*, the node identifier is the node identifier of the host message server, and the message sequence number is simply one greater than the message sequence number of the last message created by the host message server for the specified group.

The *dwTranAction* parameter controls the transaction context in which the message will be created. Typically, when multiple messages do not need to be committed atomically, *dwTranAction* will simply be set to **RIPOSTE_TRAN_ATOMIC**. In

FUJ00098220

situations where transactioning is desired, the **RIPOSTE_TRAN_START**, **RIPOSTE_TRAN_NEXT**, and **RIPOSTE_TRAN_END** transaction flags can be used to manage transactioning without needing to explicitly call RiposteStartTransaction or RiposteEndTransaction.

If a transaction is created by using the **RIPOSTE_TRAN_START** transaction flag, the calling application must either subsequently call **RiposteCreateMessageEx** using the **RIPOSTE_TRAN_END** transaction flag, or call RiposteEndTransaction or RiposteUndoTransaction to explicitly commit or abort the transaction.

Each message that is created is uniquely identified by its message identifier, which consists of the group identifier specified by *dwGroupId*, the node identifier of the host message server, and the sequence number returned in *pdwNum*. This identifier may be used subsequently to retrieve the message (see RiposteGetMessage).

**RiposteCreateMessageEx** supersedes the following functions: RiposteLogMessage, RiposteLogTestMessage, RiposteLogSystemMessage, and RiposteCreateMessage. These functions are all implemented via direct calls to **RiposteCreateMessageEx**.

To create a priority message, use RiposteCreatePriorityMessage.

Note that TRAN_HANDLE is an RPC context handle. *phTran* must specify either a pointer to a valid transaction handle, or a pointer to NULL. Even if *dwTranAction* is set to **RIPOSTE_TRAN_ATOMIC**, *phTran* cannot be NULL, it must specify a pointer to NULL.

### See Also

RiposteLogMessage, RiposteLogTestMessage, RiposteLogSystemMessage, RiposteCreateMessage, RiposteStartTransaction, RiposteEndTransaction, RiposteUndoTransaction, RiposteCreatePriorityMessage, RiposteGetMessage

Riposte32 Constants, Riposte32 Functions, Riposte32 Messaging Functions