

Broker ACI Fields

This chapter covers the following topics:

- Field Formats
- Field Descriptions

See also *Broker ACI Functions*.

Field Formats

The ACI field formats are alphanumeric, binary, or integer and include the number of bytes. For example:

Format	Description
A8, A16, A32	Alphanumeric (A-Z, 0-9, underscore, hyphen). Other characters are currently possible, but we cannot guarantee that these will work consistently across all platforms in future versions. Do not use dollar, percent, period or comma.
B16, B32	Binary
Integer (unsigned)	

The terms "null value" or "nothing specified" used for a field mean blank for alphanumeric formats and zero for integer formats.

Field Descriptions

The ACI fields are described below in alphabetical order.

ACI Field	Format	Possible Values	API Vers	I/O	Description
ADAPTER-ERROR	A8		2	O	Filled by Broker with the transport error as supplemental diagnostic data.
ADCOUNT	I4		2	O	A count of the number of times an attempt was made to deliver a UOW. The count is incremented if a UOW is backed out or timed out.
API-TYPE	B1	bits	1	I	Required for all ACI functions. See <i>API-TYPE</i> and <i>API-VERSION</i> for client and server publish and subscribe.
API-VERSION	I1	1-10	1	I	Required for all ACI functions.
BROKER-ID	A32	string	1	I	ID of the broker instance. Required for all ACI functions except <i>VERSION</i> . The <i>BROKER-ID</i> may be specified in <i>URL Style</i> or <i>Transport-method Style</i> . In order to communicate, applications must specify the same <i>BROKER-ID</i> . Note: URL style does not apply to mainframe platforms (z/OS, BS2000/OSD and z/VSE).
CLIENT-ID	I4	1-2147483647	9	O	Returns to a server application the unique instance number of a client application.
CLIENT-UID	A32	string	2	O	Applies only to client/server communication model. When a server issues a <i>RECEIVE</i> function, the user ID of the client is returned to the server in the <i>CLIENT-UID</i> field. If EntireX Security is installed, it is valid for the server application to rely on this user ID when making decisions concerning access to information. See <i>Authentication (z/OS only)</i> . Note: There is an uppercase translation when the <i>USER-ID</i> field is propagated to the <i>CLIENT-UID</i> field under EntireX Security when Broker kernel is running under z/OS.

ACI Field	Format	Possible Values	API Vers	I/O	Description
COMMITTIME	A17	YYYY MMDD HHMM SSms (milliseconds.)	7	O	Time when UOW was committed.
COMPRESLEVEL	A1	0-9 or Y N	7	I	Compression level. See <i>Data Compression</i> for client and server publish and subscribe. The following values are possible: 0 - 9 0 = no compression, 9 = maximum compression/decompression N No compression Y Compression level 6
CONV-ID	A16	string	1	I/O	A unique ID assigned to each conversation by EntireX Broker. Client and server must include the CONV-ID in their communications. Client and server can also specify the indicated textual values (capitals) in order to indicate to Broker the expected status of the conversation. Messages for the conversation are taken from the queue on a first-in, first-out basis. See <i>Conversational and Non-conversational Mode</i> . NEW On a SEND function, initiates a new conversation. On a RECEIVE function, signals readiness to receive requests for new conversations only. A CONV-ID value is assigned to the conversation, and the value is returned to the caller. OLD Applies to RECEIVE function only. Only messages for existing conversations are returned. ANY On a RECEIVE function, requests or messages are returned on a first-in, first-out basis for any conversation. On an EOC function, any conversations belonging to the caller are terminated. NONE On a SEND function, the message is non-conversational. conv-id Indicates a specific conversation. The CONV-ID value is an internally generated identifier (containing numeric characters only or alphanumeric characters) for the conversation. Application programmers are advised to make no assumptions about the contents, layout, or meaning of any part of the CONV-ID field. If the client has specified API-VERSION 3 or above, the CONV-ID contains both alphanumeric and numeric characters. If the Broker does not support UOW processing (the Broker attribute MAX-UOWS=0) or the client has specified API-VERSION or 2, the CONV-ID contains numeric characters.
CONV-STAT	I1	1 2 3	2	O	Conversation Status. See <i>Managing Conversation Contexts</i> . 1 NEW - The message is the first in a new conversation. 2 OLD - The message is part of an existing conversation. 3 NONE - The message is non-conversational.
CREDENTIALS-TYPE	I1	0 1	9	O	Determines the credentials type to be used to authenticate a user. 0 Default. Use user ID and password. 1 The token specified in the IAF token field is used.
DATA-ARCH	I1		4	I	Architecture code. For future use.
ENCRYPTION-LEVEL	I1	0 1 2	6	I	Encryption level. See <i>Encryption</i> under <i>Writing Applications using EntireX Security</i> .
ENVIRONMENT	A32	string	1	I	Information for translation user exits. The contents of the field are solely the responsibility of the application and its associated translation user exit. The field cannot be used for any other internationalization approaches and must be empty if a method other than translation user exit is used. See <i>Using the ENVIRONMENT Field with the Translation User Exit</i> for client and server publish and subscribe.
ERROR-CODE	A8		1	O	Returns an error code to the caller. The application should check the contents of this field at the completion of every Broker function. See <i>Error Handling</i> for client and server publish and subscribe. The first four digits represent the error class; the next four digits represent the error number; see also <i>Error Messages and Codes</i> .
ERRTEXT-LENGTH	I4	0-40 0-255	1 9	I	Length of the error text buffer in bytes. See <i>Call Format</i> for Assembler C COBOL Natural PL/I RPG. If there are fewer than 40 bytes, the error text may be truncated. A value of 0 (zero) means no error text. Note: In previous ACI versions, Broker kernel always returned 40 bytes of error text that were space-padded if necessary. With ACI version 9 and above, variable-length error texts can be returned to improve logging and tracing.
FORCE-LOGON	A1	Y N	6	I	Override the AUTOLOGON feature of the Broker. See AUTOLOGON. Y The attribute AUTOLOGON=YES in the Broker attribute file is overridden. See FORCE-LOGON. N Default. Use the value of the Broker attribute file for AUTOLOGON.

ACI Field	Format	Possible Values	API Vers	I/O	Description																																																
FUNCTION	I1	1-22	1	I	<p>The Broker function to be performed. A function value is required and is modified by the ACI field OPTION and the other ACI fields. See below for description of values.</p> <table border="0"> <tr> <td>1</td> <td>SEND</td> <td>14</td> <td>KERNELVERS</td> </tr> <tr> <td>4</td> <td>UNDO</td> <td>16</td> <td>SETSSLPARMS</td> </tr> <tr> <td>5</td> <td>EOC</td> <td>17</td> <td>SEND_PUBLICATION</td> </tr> <tr> <td>6</td> <td>REGISTER</td> <td>18</td> <td>RECEIVE_PUBLICATION</td> </tr> <tr> <td>7</td> <td>DEREGISTER</td> <td>19</td> <td>SUBSCRIBE</td> </tr> <tr> <td>8</td> <td>VERSION</td> <td>20</td> <td>UNSUBSCRIBE</td> </tr> <tr> <td>9</td> <td>LOGON</td> <td>21</td> <td>CONTROL_PUBLICATION</td> </tr> <tr> <td>10</td> <td>LOGOFF</td> <td>22</td> <td>REPLY_ERROR</td> </tr> <tr> <td>13</td> <td>SYNCPPOINT</td> <td></td> <td></td> </tr> </table>	1	SEND	14	KERNELVERS	4	UNDO	16	SETSSLPARMS	5	EOC	17	SEND_PUBLICATION	6	REGISTER	18	RECEIVE_PUBLICATION	7	DEREGISTER	19	SUBSCRIBE	8	VERSION	20	UNSUBSCRIBE	9	LOGON	21	CONTROL_PUBLICATION	10	LOGOFF	22	REPLY_ERROR	13	SYNCPPOINT														
1	SEND	14	KERNELVERS																																																		
4	UNDO	16	SETSSLPARMS																																																		
5	EOC	17	SEND_PUBLICATION																																																		
6	REGISTER	18	RECEIVE_PUBLICATION																																																		
7	DEREGISTER	19	SUBSCRIBE																																																		
8	VERSION	20	UNSUBSCRIBE																																																		
9	LOGON	21	CONTROL_PUBLICATION																																																		
10	LOGOFF	22	REPLY_ERROR																																																		
13	SYNCPPOINT																																																				
KERNELSECURITY	A1	Y U N	7	I/O	<p>This field is used by the application to indicate programmatically its intention to communicate with a secure/non-secure Broker. The field also indicates to the application how security has been configured for a particular Broker kernel. See Broker attribute SECURITY.</p> <p>When used as an input field, this field is used by programmer to indicate the desired security behavior of the application. If no option is specified, defaults to administrator's configuration setup.</p> <table border="0"> <tr> <td>Y</td> <td>EntireX Security</td> </tr> <tr> <td>U</td> <td>User-written Security</td> </tr> <tr> <td>N</td> <td>No security</td> </tr> </table> <p>Notes</p> <ul style="list-style-type: none"> • Output In version 7 or above, this field returns the output value when executing the KERNELVERSION command. • Input In version 8 or above, the application can programmatically specify the desired security behavior for all commands other than KERNELVERSION. 	Y	EntireX Security	U	User-written Security	N	No security																																										
Y	EntireX Security																																																				
U	User-written Security																																																				
N	No security																																																				
LOCALE-STRING	A40	string	4	I	<p>It is assumed that you have read the document <i>Internationalization with EntireX</i> and are familiar with the various internationalization approaches described there.</p> <p>The locale string tells the broker the encoding of the data. No conversion is done within the broker's stub. The application must ensure the data provided matches the locale string. The locale string is case-insensitive, also dashes '-' and underscores '_' are ignored (dashes and underscore improve human readability). See <i>Using Internationalization</i> for client and server publish and subscribe.</p>																																																
LOG-COMMAND	I1	0 1	9	I	<p>Components that communicate with Broker can trigger command logging by setting this field. By default, command logging is based on the command log filters set in the kernel. You may override these kernel settings programmatically by setting this LOG-COMMAND field. If this field is set, all associated commands will be logged.</p> <p>Note: If command logging is not enabled for your kernel, you must first contact your administrator.</p>																																																
LONG-BROKER-ID-LENGTH	I4	0-2147483647	10	I/O	<p>Length of LONG-BROKER-ID. If the value is non-zero, specify the value of LONG-BROKER-ID directly after the ACI control block. The LONG-BROKER-ID overrides any BROKER-ID value.</p> <p>With the LONG-BROKER-ID you can now specify numeric IPv6 addresses. Some sample values:</p> <pre>tcpip: / [[2001:0db8:85a3:08d3:1319:8a2e:–0370:–7347] : 3930] : 3930</pre> <pre>[2001:0db8:85a3:08d3:1319:8a2e:–0370:–7347] : 3930:TCP</pre> <pre>(2001:0db8:85a3:08d3:1319:8a2e:–0370:–7347) : 3930:TCP</pre> <p>The IP address is enclosed in square brackets or parentheses.</p>																																																
MSG-ID	B32		2	I/O	Not used by EntireX Broker.																																																
MSG-TYPE	A16		2	I/O	Not used by EntireX Broker.																																																
NEWPASSWORD	B32	Can contain binary data.	2	I	<p>Specifies a new password to be transmitted to the Broker kernel to check the authentication of the application. See <i>Authentication</i>.</p> <p>The current password can be changed only when the client or server authenticates itself. This occurs on the first Broker ACI function (can be LOGON) and requires the application to assign to the Broker ACI fields PASSWORD and NEWPASSWORD.</p>																																																
OPTION	I1	0-21	1	I	<p>Provides additional information that modifies the behavior of the Broker ACI functions.</p> <table border="0"> <tr> <td>0</td> <td>no option</td> <td>8</td> <td>NEXT</td> <td>16</td> <td>QUERY</td> </tr> <tr> <td>1</td> <td>MSG</td> <td>9</td> <td>PREVIEW</td> <td>17</td> <td>SETSTATUS</td> </tr> <tr> <td>2</td> <td>HOLD</td> <td>10</td> <td>COMMIT</td> <td>18</td> <td>ANY</td> </tr> <tr> <td>3</td> <td>IMMED</td> <td>11</td> <td>BACKOUT</td> <td>19</td> <td>reserved for future use</td> </tr> <tr> <td>4</td> <td>QUIESCE</td> <td>12</td> <td>SYNC</td> <td>20</td> <td>DURABLE</td> </tr> <tr> <td>5</td> <td>EOC</td> <td>13</td> <td>ATTACH</td> <td>21</td> <td>CHECKSERVICE</td> </tr> <tr> <td>6</td> <td>CANCEL</td> <td>14</td> <td>DELETE</td> <td></td> <td></td> </tr> <tr> <td>7</td> <td>LAST</td> <td>15</td> <td>EOCCANCEL</td> <td></td> <td></td> </tr> </table>	0	no option	8	NEXT	16	QUERY	1	MSG	9	PREVIEW	17	SETSTATUS	2	HOLD	10	COMMIT	18	ANY	3	IMMED	11	BACKOUT	19	reserved for future use	4	QUIESCE	12	SYNC	20	DURABLE	5	EOC	13	ATTACH	21	CHECKSERVICE	6	CANCEL	14	DELETE			7	LAST	15	EOCCANCEL		
0	no option	8	NEXT	16	QUERY																																																
1	MSG	9	PREVIEW	17	SETSTATUS																																																
2	HOLD	10	COMMIT	18	ANY																																																
3	IMMED	11	BACKOUT	19	reserved for future use																																																
4	QUIESCE	12	SYNC	20	DURABLE																																																
5	EOC	13	ATTACH	21	CHECKSERVICE																																																
6	CANCEL	14	DELETE																																																		
7	LAST	15	EOCCANCEL																																																		

ACI Field	Format	Possible Values	API Vers	I/O	Description
PARTNER-BROKER-ID	A32	string	9	O	ID of the partner broker. Deprecated.
PASSWORD	A32	Can contain binary data.	1	I	Specifies a password to be transmitted to the Broker to check the authentication of the application. See <i>Authentication</i> and <i>FORCE-LOGON</i> .
PTIME	A8		2	I	Not used by EntireX Broker.
PUBLICATION-ID	A16	string, case-sensitive.	8	I/O	<p>Publication ID for publish-and-subscribe communication model.</p> <p>A unique ID assigned to each publication by EntireX Broker. Publisher and subscribers must include the publication ID and the <i>CONV-ID</i> in their communications. Publisher and subscriber can also specify the indicated textual value (capitals) in order to indicate to Broker the expected status of the publication. Messages for the publication are queued to the topic on a first-in, first-out basis.</p> <p>NEW On a <i>SEND_PUBLICATION</i> function, initiates a new publication. On a <i>RECEIVE_PUBLICATION</i> function, signals readiness to obtain next available publication. A publication ID value is assigned to the publication, and the value is returned to the caller.</p> <p>publication-id Indicates a specific publication. The <i>PUBLICATION-ID</i> value is an internally generated identifier (containing alphanumeric characters) for the publication. Application programmers are advised to make no assumptions about the content, layout or meaning of any part of the <i>PUBLICATION-ID</i> field.</p>
RECEIVE-LENGTH	B32	Binary.	1	I/O	<p>Specifies the length of receive buffer, in bytes. The maximum length depends on the transport method:</p> <p>NET 30,545 TCP 2,147,483,647 SSL 2,147,483,647</p> <p>Note: Under z/OS with Adabas version 8, the value for NET is the same as for TCP and SSL.</p> <p>See <i>Using Send and Receive Buffers</i> for client and server publish and subscribe.</p>
RETURN-LENGTH	I4		1	O	<p>Length, in bytes, of the data returned.</p> <p>See <i>Using Send and Receive Buffers</i> for client and server publish and subscribe.</p>
SECURITY-TOKEN	B32	binary	1	I/O	<p>The contents of this field depend heavily on the implementation of the security exits.</p> <p>This field is utilized by EntireX Security. The application must maintain <i>SECURITY-TOKEN</i> between commands and not change this value. We recommend that the application allocate a separate ACI control block for each user if it issues commands on behalf of more than one user. For objects executing inside Web servers, assigning a unique value, such as 'session ID', to the <i>ACI TOKEN</i> field is highly recommended to ensure uniqueness of user at same physical location. See <i>Ignore Security Token</i>.</p> <p>If EntireX Security is not implemented, and you choose to write your own security exits your can transmit an initial value to your security exit as a credential that is used to calculate the actual security token. After an application's authenticity has been verified by the security exits, the <i>SECURITY-TOKEN</i> can be used to avoid additional authentication checks.</p>
SEND-LENGTH	B32	binary	1	I/O	<p>Specifies the length of data being sent, in bytes. The maximum length depends on the transport method:</p> <p>NET 30,545 TCP 2,147,483,647 SSL 2,147,483,647</p> <p>Note: Under z/OS with Adabas version 8, the value for NET is the same as for TCP and SSL.</p> <p>See <i>Using Send and Receive Buffers</i> for client and server publish and subscribe.</p>
SERVER-CLASS SERVER-NAME SERVICE	A32 each	string, case-sensitive	1	I/O	<p>A client uses these fields to identify the service that it requires. A server uses this field to offer a service.</p> <p>Using all three fields allows you to organize servers, making them easier to identify, monitor, and maintain. Servers can be organized into server-classes, with each server providing a number of different services. Each service must be defined in the attribute file (see <i>Service-specific Attributes</i> under <i>Broker Attributes</i>).</p> <p>The service fields are required with <i>SEND</i>, <i>RECEIVE</i>, and <i>EOC</i> functions when <i>CONV-ID</i> is set to <i>NEW</i>, <i>OLD</i>, or <i>ANY</i>. When a <i>CONV-ID</i> is supplied, the service fields are ignored.</p> <p><i>SERVICE=* or SERVER-NAME=*</i> can be used on a <i>RECEIVE</i> function to indicate all services within a specified server or all servers within a specified server class.</p> <p>Note: Server classes "SAG", "Entire", "Adabas", "Natural", "ETB", "RPC" and Broker are reserved for Software AG. Do not use them in your applications.</p>
STATUS	I1		2	I/O	Not used by EntireX Broker.
STORE	I1	0 1 2	2	I/O	<p>Persistence or non-persistence of a UOW. Used with the first <i>SEND</i> function for a UOW to specify whether the UOW is persistent or not. Once established, the persistence of a UOW cannot be altered.</p> <p>0 none - Defaults to the value specified for the service. 1 OFF - The UOW is not persistent. 2 BROKER - The UOW is persistent.</p>

ACI Field	Format	Possible Values	API Vers	I/O	Description
TOKEN	A32	string, case-sensitive.	1	I	<p>Optionally identifies the caller and, when used, is required for all Broker ACI functions except <code>VERSION</code>. See <code>USER-ID</code> and <code>TOKEN</code> for client and server publish and subscribe.</p> <p>Warning: USER-ID and TOKEN must be specified by all applications that use UOWs held in the persistent store, and by all publisher and subscriber applications where publication and subscription data is held in the persistent store.</p>
TOPIC	A96	string, case-sensitive	8	I/O	<p>Topic name for publish and subscribe communication model.</p> <p>A publisher uses this field to identify the topic name required. A subscriber uses this field to indicate the topic from which publications are to be obtained. Each topic must be defined in the attribute file. See <i>Topic-specific Attributes</i> under <i>Broker Attributes</i>.</p>
UOWID	A16		3	I/O	<p>A unique identifier for a UOW.</p> <p>The value is returned on the first <code>SEND</code> or <code>RECEIVE</code> command within a UOW; the value must be provided on all subsequent <code>SEND</code>, <code>RECEIVE</code> and <code>SYNCPOINT</code> commands related to the same UOW. Client and server can also specify the indicated textual value (capitals) in order to indicate to Broker the following:</p> <p>BOTH Since a server receives a UOW and replies with a different UOW, both UOWs can be committed or backed out by specifying <code>UOWID=BOTH</code> for the <code>SYNCPOINT</code> command.</p> <p>uowid The <i>uowid</i> must be supplied in subsequent <code>SEND</code>, <code>RECEIVE</code> and <code>SYNCPOINT</code> commands related to the same UOW.</p>
UOWSTATUS	I1		3	O	<p>Contains the status of a UOW. EntireX Broker returns the <code>UOWSTATUS</code> field to the calling application in order to provide information about the condition of the specified UOW.</p> <p>1 RECEIVED - One or more messages have been sent as part of a UOW but the UOW has not yet been committed. 2 ACCEPTED - The UOW has been committed by the sender. 3 DELIVERED - The UOW is currently being received. 4 BACKEDOUT - The UOW has been backed out by the sender. 5 PROCESSED - The UOW has been received and the receiver has committed it. 6 CANCELLED - The UOW has been cancelled by the receiver. 7 TIMEOUT - The UOW was not processed within the time allowed. 8 DISCARDED - The UOW was not persistent and its data was discarded as the result of a restart.</p> <p>With the exception of <code>DELIVERED</code>, all <code>UOWSTATUS</code> values are persistent. Persistent values are kept until they are explicitly deleted by the user or the time limit is exceeded. The lifetime of the <code>UOWSTATUS</code> value is determined by the broker attribute <code>UWSTATP</code>.</p> <p><code>UOWSTATUS</code> values in the following table are returned on a <code>RECEIVE</code> function to indicate whether the message being transferred is part of a UOW and, if so, its sequence within the UOW:</p> <p>0 NONE - The message is not part of a UOW. 9 FIRST - The message is the first message in a UOW. 10 MIDDLE - The message is neither the first nor the last in the UOW. 11 LAST - The message is the last message in the UOW. 12 ONLY - The message is the only message in the UOW.</p>
UOW-STATUS-PERSIST		0 - 255	3	I	<p>The value of the <code>UOW-STATUS-PERSIST</code> field is used as a multiplier to calculate the lifetime for the persistent status of a UOW. The value is multiplied by the value of the broker attribute <code>UWTIME</code>. The value 255 can be specified to indicate no persistent status.</p> <p>0 Means that the multiplier will have the same value as the <code>UWSTATP</code> Broker attribute. 255 Means that there will be no persistent status for UOWs. 1-254 Any number in this range is a valid multiplier.</p>
USER-DATA	B16	binary	2	I/O	<p>Conversation User Data. See <i>Managing Conversation Contexts</i>.</p>
USER-ID	A32	string, case-sensitive	1	I	<p>Identifies the caller and is required for all Broker ACI functions except <code>VERSION</code>. See <code>USER-ID</code> and <code>TOKEN</code> for client and server publish and subscribe.</p> <p>Warning: USER-ID and TOKEN must be specified by all publisher and subscriber applications where publication and subscription data is held in the persistent store.</p>
USTATUS	A32	string	3	I/O	<p>User-defined information about a unit of work (UOW). It can be transmitted on a <code>SEND</code>, <code>RECEIVE</code>, or <code>SYNCPOINT</code> function and is returned to applications that query the status of the UOW. To update the <code>USTATUS</code> field, use <code>OPTION=SETUSTATUS</code>.</p>

ACI Field	Format	Possible Values	API Vers	I/O	Description
UWSTAT-LIFETIME	A8	nS nM nH nD	8	I	<p>Add value for persistent status lifetime in client and server communication model.</p> <p>This field is used to calculate the lifetime of the UOW status. The value of this field determines how long the UOW status is to be retained in the persistent store after the UOW is processed or timed out if it is not processed. This is an alternative to specifying UOW-STATUS-PERSIST.</p> <p>UWSTAT-LIFETIME is specified on the first SEND_PUBLICATION function for a UOW; it is not allowed on a RECEIVE_PUBLICATION function.</p> <p>nS The number of additional seconds the UOW status will exist.</p> <p>nM The number of additional minutes the UOW status will exist.</p> <p>nH The number of additional hours the UOW status will exist.</p> <p>nD The number of additional days the UOW status will exist.</p>
UWTIME	A8	nS nM nH nD	3	I	<p>The lifetime of a UOW. The UOW exists until its lifetime expires or it is explicitly cancelled or backed out with SYNCPOINT OPTION=CANCEL or SYNCPOINT OPTION=BACKOUT.</p> <p>If the UOW is not committed, backed out, or cancelled before its UWTIME expires, the UOW is discarded and its status becomes TIMEOUT.</p> <p>UWTIME is specified on the first SEND function for a UOW; it is not allowed on a RECEIVE function.</p> <p>nS The number of seconds the UOW can exist.</p> <p>nM The number of minutes the UOW can exist.</p> <p>nH The number of hours the UOW can exist.</p> <p>nD The number of days the UOW can exist.</p>
VARLIST-OFFSET	I4	0-2147483647	10	I	For Software AG internal use only.
WAIT	A8	NO YES nS nM nH	1	I	<p>When a WAIT value (other than NO) is specified on a SEND or RECEIVE function, the caller will wait for a reply until the message is received or the specified time limit has been reached. See <i>Blocked and Non-blocked Broker Calls</i> for client and server publish and subscribe.</p> <p>NO Default. No wait. Control is returned to the caller.</p> <p>nS The number of seconds the caller will wait for a reply.</p> <p>nM The number of minutes the caller will wait for a reply.</p> <p>nH The number of hours the caller will wait for a reply.</p> <p>YES Depending on the role of the user (client, server, publisher or subscriber), the respective attribute is used (CLIENT-NONACT SERVER-NONACT PUBLISHER-NONACT SUBSCRIBER-NONACT). If a server registers multiple services, the highest value of all the services registered is taken as wait time for the server. However, if the user is both client and server, CLIENT-NONACT is also used for calculating the wait time.</p> <p>All different roles provide non-activity attributes. The maximum value is taken for the wait time.</p>