Directory Server Target Entries

Software AG communication information for your product is stored in one or more Software AG Directory Servers. The client’s send message includes the target server name. Your Software AG product forwards the name and a use qualifier to the Directory Server, which returns an appropriate qualified URL (Universal Resource Locator) for the target back to your product.

Physical connection information (transport protocol , protocol specific parameters, timeout, and so on) must be entered in Directory Server target entries as qualified URLs before this communication can occur. The qualified URL contains the information required to direct the message to the correct target. The qualifier identifies which target URL is to be returned, based on the use implied by the qualifier. For example, a client *send* request returns an *access* target URL .

Directory Server target entries can be added manually using the System Management Hub. For more information, read *Maintaining Targets*.

This chapter covers the following topics:

- Qualified URL Structure
- Qualifiers
- Protocols
- Parameters

Qualified URL Structure

Physical connection information (transport protocol , protocol specific parameters, timeout, and so on) must be entered in the Directory Server target entries as qualified URLs before the Directory Server can be used for Software AG communication. Each qualified URL is specified in this format:

qualifier`.`*protocol*`://`*host*`:`*port*`[?`*parm*`=`*value*`][&`*parm*`=`*value*`]...`

For example:

```
access.tcpip://serverhost:3001?retry=3
```

Entry	Meaning
"*qualifier*"	The use of this target URL. Three types of qualifiers are supported: "access", "connect", and "listen". For more information, read *Qualifiers*.
"*protocol*"	The communication protocol that will be used to connect to the server. For more information, read *Protocols*.
"*host*"	The name of the host computer where the server runs.
"*port*"	The server's port. The port is a destination or a receiving port, depending upon URL usage. Refer to the documentation for the specific server application to identify its valid port numbers and how they are assigned..
"*parm*"	One of multiple optional parameters that can be used. The first parameter is preceded by a "?" and subsequent parameters, if any, are preceded by an "&". For more information, read *Parameters*.
"*value*"	The value of the parameter.

Qualifiers

URLs are qualified in the Directory Server target entries by their use. Qualifiers are used to specify this use. Three qualifiers (uses) of a URL are supported in the Software AG Directory Server, as described in the following table:

Qualifier (Use)	Description
access	Defines a communication path between the client and the server. The path provides the means for the client to communicate with the server either directly or through a proxy; this communication path tells the client where to find the server. Internally, a URL with this specification appears as an "XTSaccess" URL.
listen	Defines a listen port for the server or the proxy. Internally, a URL with this specification appears as an "XTSlisten" URL.
connect	Defines an active connection between a server and a proxy or between a proxy and an Entire Net-Work node. Internally, a URL with this specification appears as an "XTSconnect" URL.

Protocols

The following communication protocols can be used in Directory Server URLs.

Protocol	Description
HTTP11	Although this protocol is still listed on Directory Server administration screens in the System Management Hub, this protocol is no longer supported.
MHDR	Only Software AG products that require the proxy can use this protocol. The MHDR protocol allows the proxy to communicate with these Software AG products. The MHDR protocol supports two-byte database IDs; therefore, databases with database IDs greater than "255" can be accessed using this protocol.
RDA	Only Software AG products that require the proxy can use this protocol. The RDA protocol allows the proxy to communicate with these Software AG products. The RDA protocol does not support two-byte database IDs; therefore access is limited to database IDs less than "256".
SSL	The SSL (Secure Sockets Layer) protocol enables secure TCP/IP point-to-point connections. **Note:** A random file is required on UNIX systems if the SSL protocol is used or errors will occur. For complete information, read *SSL Random File Requirements on UNIX Systems*.
TCP/IP	The TCP/IP protocol is the standard communication protocol used. It provides the most basic and efficient service.

Parameters

The parameters you can specify in a qualified URL vary, depending on the protocol and qualifier selected. The following table describes the parameters available and indicates which protocols and qualifiers support them.

Parameter	Qualifier Support	Protocol Support	Description
`cafile`	access connect listen (client authentication only)	SSL - C applications only	Identifies the file containing the trusted CA certificates. The CA’s certificate that signed an inbound certificate must reside in this file. **Note:** The file name specified may include the path information, unless a value for parameter `capath` is specified. The `cafile` and `capath` parameters are required for client and server authentication.

Parameter	Qualifier Support	Protocol Support	Description
`capath`	access connect listen (client authentication only)	SSL - C applications only	Supplies a hash value generated by the OpenSSL tool that specifies the location of a `cafile` in a complex CA structure. This location is not a path. If parameter `cafile` includes location information, the value of `capath` should be ".", which is also the `capath` default. The `cafile` and `capath` parameters are required for client and server authentication.
`cert_file`	access (client authentication only) connect listen	SSL - C applications only	Specifies the file containing the participant's certificate. The certificate file may contain the participant's private key. **Note:** The file name specified may include the path information. This is useful if the certificate is not in the current directory.
`cert_passwd`	access (client authentication only) connect listen	SSL - C applications only	Specifies the password for extracting information from the certificate file. **Note:** You can specify a fully qualified file name for this parameter. In this case, the file name you provide must contain the password.
`charset`	all	RDA	Identifies the character encoding of the classic Entire Net-Work node associated with the URL. The value "EBCDIC" must be specified when and only when the URL is for a mainframe connection; no other value can be specified. The default value is "ASCII" which applies to non-mainframe connections.
`chirpinterval`	all	RDA SSL TCP/IP	Specifies the number of seconds to wait between chirp attempts for this connection. Chirping is the communication mechanism used to validate the availability of the connection specified by the URL. The valid range is "0" through the maximum integer that can be stored by your operating system. The default value is "300"seconds (5 minutes). A value of "0" implies the default, "300".

Parameter	Qualifier Support	Protocol Support	Description
key_file	all	SSL - C applications only	Specifies the file containing the server's private key. Must be specified if the private key is kept separate from the certificate file. **Note:** The file name specified may include the path information. This is useful if the certificate is not in the current directory.
keystore	access (client authentication only) connect listen	SSL - Java application only	Identifies the Java keystore containing the participant's certificate and private key.
keystore_passwd	access (client authentication only) connect listen	SSL - Java application only	Specifies the password for extracting information from keystore.
node	all	RDA	Specifies the node ID by which this node will be known to a classic Entire Net-Work installation. The valid range is 1 through 65535. The default value is "7654". If more than one proxy is connected in the same Entire Net-Work domain, the node and nodename must be given to avoid conflicts.
nodename	all	RDA	Specifies the node name by which this node will be known to a classic Entire Net-Work installation. The default value is the name of the proxy. If more than one proxy is connected in the same Entire Net-Work domain, the node and nodename must be given to avoid conflicts.
priority	---	none	Reserved for future use.
random_file	all	SSL - C applications only	Identifies a text file that contains at least 14 random characters. The random characters in this file are used by the encryption routines to ensure that encryption itself occurs in a random manner.

Parameter	Qualifier Support	Protocol Support	Description
`raw`	all	RDA SSL TCP/IP	Indicates whether transport subsystem headers are sent. If present, then no transport subsystem headers are sent and no proxy is possible. Values are "on" and "off". The default value is "off". RDA target entries must specify `raw=on` or the connections will not work.
`reconnect`	all	RDA SSL TCP/IP	Indicates whether or not to reconnect if disconnected. Values are "on" or "off". The default value is "on".
`recvtimeout`	all	RDA SSL TCP/IP	Specifies a protocol timeout value in seconds. Valid values range from "0" through the maximum integer that can be stored by your operating system. The default is "60" seconds. A value of "0" implies the default, "60". This parameter is most useful for performance tuning. We do not recommend that you modify this parameter unless necessary. For assistance, contact Software AG Customer Support.
`retry`	all	RDA SSL TCP/IP	Specifies the number of times to retry a connection. The valid range is 0 through 2147483648. The default value is "0" (no retry).
`retryint`	all	RDA SSL TCP/IP	Specifies the interval in seconds between retries. The valid range is 0 through 2147483648. The default value is "60000" seconds.
`security`	all	RDA	Specifies the name of a security file containing a list of IP addresses authorized to access this protocol. There is no default value.

Parameter	Qualifier Support	Protocol Support	Description
`sendtimeout`	all	RDA SSL TCP/IP	Specifies a protocol timeout value in seconds. Valid values range from "0" through the maximum integer that can be stored by your operating system. The default is "60" seconds. A value of "0" implies the default, "60". This parameter is most useful for performance tuning. We do not recommend that you modify this parameter unless necessary. For assistance, contact Software AG Customer Support.
`trace`	all	RDA SSL TCP/IP	Indicates whether or not to trace this connection. Values are "on" or "off". The default value is "off".
`truststore`	access connect listen (client authentication only)	SSL - Java application only	Identifies the Java truststore containing the trusted CA certificates. The CA’s certificate that signed an inbound certificate must reside in this file.
`truststore_passwd`	access connect listen (client authentication only)	SSL - Java application only	Specifies the password for extracting information from the truststore.
`ttl`	---	none	Reserved for future use.

Parameter	Qualifier Support	Protocol Support	Description
`verify`	access connect listen (client authentication only)	SSL - both C and Java applications	Identifies the certificate processing level. For C applications, valid values are: 0 (No peer verification occurs. This is the default value.) 1 (The application requests that the peer certificate be verified.) 2 (The application requests that the peer certificate be verified. A fatal condition occurs if there is no certificate.) 4 (The application requests that the peer certificate be verified only once.) 8 (The application requests that the issuer name is checked against the host name.) Values 1, 2, and 4 can be specified in combination. For example, if you want to specify both 1 and 2, you would add them and set the `verify` parameter to "3". **Note:** This parameter must be set to "3" if you are performing client authentication. For Java applications, valid values are: 0 (No peer verification occurs. This is the default value.) 1 (The application requests that the peer certificate be verified.) 2 (The application requests that the peer certificate be verified. A fatal condition occurs if there is no certificate.) Values 4 and 8 are not valid for Java.
`version`	all	SSL - both C and Java applications	Indicates the SSL version: 1 (TLSv1) 2 (SSLv2). This value is required for Java applications. 3 (SSLv23). For C applications only, this indicates that Version 2 or 3 should be used. 4 (SSLv3)