Setting up Broker Instances

- Starting and Stopping the Default Broker
- Running Broker with SSL or TLS Transport
- Uniqueness Test for Broker ID
- Tracing EntireX Broker
- Protecting a Broker against Denial-of-Service Attacks

Starting and Stopping the Default Broker

If check box **Turn on Autostart for default EntireX Broker** is checked during installation, the default broker ETB001 is started.

To start the default broker

- From the Windows start menu, choose **Software AG > Start Servers > Start EntireX Default Broker**.

To stop the default broker

- From the Windows start menu, choose **Software AG > Stop Servers > Stop EntireX Default Broker**.

Running Broker with SSL or TLS Transport

Before starting the Broker, it must be configured to correctly use SSL or TLS as a transport mechanism:

- Step 1: Modify Broker-specific Attributes
- Step 2: Modify SSL-specific Attributes

Step 1: Modify Broker-specific Attributes

Append `"-SSL"` to the `TRANSPORT` attribute. For example:

```
DEFAULTS = BROKER
TRANSPORT = TCP-SSL
```

See also `TRANSPORT`.

Step 2: Modify SSL-specific Attributes

Set the SSL or TLS attributes, for example:

```
DEFAULTS = SSL
 KEY-STORE = "C:\SoftwareAG\EntireX\etc\ExxAppCert.pem"
 KEY-PASSWD-ENCRYPTED = MyAppKey
 KEY-FILE = "C:\SoftwareAG\EntireX\etc\ExxAppKey.pem"
 VERIFY-CLIENT = N
 PORT=1958
```

where *1958* is the default but can be changed to any port number.

See also *SSL-specific Attributes* under *Broker Attributes* and *SSL or TLS and Certificates with EntireX*.

Uniqueness Test for Broker ID

To guarantee that a broker ID is unique on one machine, a named semaphore is created at initialization. If this semaphore already exists for this broker ID, initialization is terminated with message ETBE0168, "This instance of broker already running". If as a result of an abnormal broker termination this semaphore cannot be deleted completely, you can force a restart of the Broker with Broker attribute `FORCE=YES`.

Tracing EntireX Broker

This section covers the following topics:

- Broker TRACE-LEVEL Attribute
- Attribute File Trace Setting
- Deferred Tracing

Broker TRACE-LEVEL Attribute

The Broker `TRACE-LEVEL` attribute determines the level of tracing to be performed while Broker is running. The Broker has a master `TRACE-LEVEL` specified in the Broker section of the attribute file as well as several individual `TRACE-LEVEL` settings that are specified in the following sections of the attribute file. You can also modify the different `TRACE-LEVEL` values while Broker is running, without having to restart the Broker kernel for the change to take effect.

For temporary changes to `TRACE-LEVEL` without restarting the Broker, use the System Management Hub.

Individual Settings	Specified in Attribute File Section
Master trace level	`DEFAULTS=BROKER`
Persistent store trace level	`DEFAULTS=ADABAS | CTREE | DIV` (currently not available for DIV)
Conversion trace level	Trace option of the `CONVERSION` parameter that can be defined in `DEFAULTS=SERVICE | TOPIC`
Security trace level	`DEFAULTS=SECURITY`

These individual TRACE-LEVEL values determine the level of tracing within each subcomponent. If not specified, the master TRACE-LEVEL is used.

Attribute File Trace Setting

Trace Level	Description
0	No tracing. Default value.
1	Traces incoming requests, outgoing replies, and resource usage.
2	All of Trace Level 1, plus all main routines executed.
3	All of Trace Level 2, plus all routines executed.
4	All of Trace Level 3, plus Broker ACI control block displays.
8	All of Trace Level 4, plus Adabas Persistent Store Adabas control blocks.

Note:
Trace levels 2 and above should be used only when requested by Software AG support.

Deferred Tracing

It is not always convenient to run with TRACE-LEVEL defined, especially when higher trace levels are involved. Deferred tracing is triggered when a specific condition occurs, such as an ACI response code or a broker subtask abend. Such conditions cause the contents of the trace buffer to be written, showing trace information leading up the specified event. If the specified event does not occur, the Broker trace will contain only startup and shutdown information (equivalent to TRACE-LEVEL=0). Operating the trace in this mode requires the following additional attributes in the broker section of the attribute file. Values for TRBUFNUM and TRAP-ERROR are only examples.

Attribute	Value	Description
TRBUFNUM	3	Specifies the deferred trace buffer size = 3 * 64 K.
TRMODE	WRAP	Indicates trace is not written until an event occurs.
TRAP-ERROR	322	Assigns the event ACI response code 00780322 "PSI: UPDATE failed".

Protecting a Broker against Denial-of-Service Attacks

An optional feature of EntireX Broker is available to protect a broker running with SECURITY=YES against denial-of-service attacks. An application that is running with invalid user credentials will get a security response code. However, if the process is doing this in a processing loop, the whole system could be affected. If PARTICIPANT-BLACKLIST is set to YES, EntireX Broker maintains a blacklist to handle such "attacks". If an application causes ten consecutive security class error codes within 30 seconds, the blacklist handler puts the participant on the blacklist. All subsequent requests from this participant are blocked until the BLACKLIST-PENALTY-TIME has elapsed.

Server Shutdown Use Case

Here is a scenario illustrating another use of this feature that is not security-related.

An RPC server is to be shut down immediately, using Broker Command and Information Services (CIS), and has no active request in the broker. The shutdown results in the `LOGOFF` of the server. The next request that the server receives will probably result in message 00020002 "User does not exist", which will cause the server to reinitialize itself. It was not possible to inform the server that shutdown was meant to be performed.

With the *blacklist*, this is now possible. As long as the blacklist is not switched off, when a server is shut down immediately using CIS and when there is no active request in the broker, a marker is set in the blacklist. When the next request is received, this marker results in message 00100050 "Shutdown IMMED required", which means that the server is always informed of the shutdown.