Tracing webMethods EntireX

This chapter describes the various techniques available for troubleshooting, tracing and logging with EntireX components.
It covers the following topics:

- Table Summarizing Tracing for webMethods EntireX Components
- Tracing EntireX Broker
- Tracing Broker Agent
- Tracing Broker Stubs
- Tracing Enterprise JavaBeans
- Logging Enterprise JavaBeans
- Tracing EntireX Java ACI
- Tracing Java RPC Server
- Tracing the RPC Runtime
- Tracing the RPC Server
- Tracing the XML/SOAP Runtime
- Tracing the EntireX RPC-ACI Bridge

Table Summarizing Tracing for webMethods EntireX Components

EntireX Component	Use Tracing Technique for	Tracing Technique
Broker ActiveX Control	Transport-related problems Requests to, replies from the Broker or Broker Agent	*Tracing Broker Stubs*
EntireX Broker ACI under Windows	Transport-related problems Requests to, replies from the Broker or Broker Agent	*Tracing Broker Stubs*
EntireX Broker Agent	Transport-related problems Requests to, replies from the Broker or Broker Agent	*Tracing Broker Agent*
EntireX Broker under Windows	Processing within the Broker Requests to, replies from clients/server	*Tracing EntireX Broker*

EntireX Component	Use Tracing Technique for	Tracing Technique
DCOM Wrapper	Transport-related problems Requests to, replies from the Broker or Broker Agent	*Tracing Broker Stubs*
	RPC-related problems on the client side Requests to, replies from RPC Servers Requests to, replies from the Broker	*Tracing the RPC Runtime*
EntireX Java ACI	Transport-related problems Requests to, replies from the Broker or Broker Agent	*Tracing EntireX Java ACI*
Java Wrapper	Transport-related problems Requests to, replies from the Broker or Broker Agent	*Tracing EntireX Java ACI*
Wrapper for EJB	Transport-related problems Requests to, replies from the Broker or Broker Agent	*Tracing Enterprise JavaBeans*
	Log information to the application server the JavaBean is executing in	*Logging Enterprise JavaBeans*
EntireX Java RPC Server	Transport-related problems Requests to, replies from the Broker or Broker Agent	*Tracing Java RPC Server*
EntireX IDL Tester		
.NET Wrapper	Transport-related problems Requests to, replies from the Broker or Broker Agent	*Tracing Broker Stubs*
	RPC-related problems on the client side Requests to, replies from RPC servers Requests to, replies from the Broker	*Tracing the RPC Runtime*
C Wrapper	Transport-related problems Requests to, replies from the Broker or Broker Agent	*Tracing Broker Stubs*
	RPC-related problems on the client side Requests to, replies from RPC servers Requests to, replies from the Broker	*Tracing the RPC Runtime*
EntireX RPC Server under Windows	RPC-related problems on the server side Requests to, replies from RPC clients Requests to, replies from the Broker	*Tracing the RPC Server*
	Transport-related problems Requests to, replies from the Broker or Broker Agent	*Tracing Broker Stubs*
Broker HTTP(S) Agent	Transport-related problems Requests to, replies from the Broker or Broker Agent	*Tracing EntireX Java ACI*

EntireX Component	Use Tracing Technique for	Tracing Technique
EntireX XML/SOAP RPC Server	For XML/SOAP RPC Server-related problems.	*Tracing the XML/SOAP Runtime*
	Transport-related problems Requests to, replies from the Broker or Broker Agent	*Tracing EntireX Java ACI*
EntireX XML Tester		
EntireX XML/SOAP Listener	For XML/SOAP Listener-related problems.	*Tracing the XML/SOAP Runtime*
	Transport-related problems Requests to, replies from the Broker or Broker Agent	*Tracing EntireX Java ACI*
XML/SOAP Wrapper	For XML/SOAP Wrapper-related problems.	*Tracing the XML/SOAP Runtime*
	Transport-related problems Requests to, replies from the Broker or Broker Agent	*Tracing EntireX Java ACI*
EntireX RPC-ACI Bridge		*Tracing the EntireX RPC-ACI Bridge*

Tracing EntireX Broker

To switch on tracing

- Set the attribute `TRACE-LEVEL` in the broker attribute file
  - for minimal trace output to "1"
  - for detailed trace output to "2"
  - for full trace output to "3"

  Example:

  ```
  TRACE-LEVEL=2
  ```

To switch off tracing

- Set the attribute `TRACE-LEVEL` in the broker attribute file to 0:

  ```
  TRACE-LEVEL=0
  ```

  Or:
  Omit the `TRACE-LEVEL` attribute.

To display the trace file (under Windows)

- In System Management Hub, select EntireX *n.n.n*, then EntireX Broker, then the Broker ID you are interested in, then choose **Show Log File**.

Trace Output

The trace file, *BrokerID.LOG*, is written to the *Broker Directory* directory.

Related Information

EntireX Broker Return Codes

Tracing Broker Agent

To switch on tracing

- Set the parameter Trace Option to ON. For the complete table of parameters, see *Using the SSL Agent* and *Using the TCP Agent*.

To switch off tracing

- Set the parameter Trace Option to OFF.

  Or:
  Omit the parameter Trace Option.

Trace Output

The trace output will be written to STDOUT.

If the Broker Agent is started with the System Management Hub, the trace output is written to the subfolder *etc* of the EntireX main directory. The file name is *BrokerAgent.<agent name>.log*.

Tracing Broker Stubs

The broker stubs provide an option for writing trace files.

To switch on tracing for the broker stub

- Before starting the client application, set the environment variable `ETB_STUBLOG`:

Trace Value	Trace Level	Description
0	NONE	No tracing.
1	STANDARD	Traces initialization, errors, and all ACI request/reply strings.
2	ADVANCED	Used primarily by system engineers, traces everything from level 1 and provides additional information - for example the Broker ACI control block - as well as transport information.
3	SUPPORT	This is full tracing through the stub, including detailed traces of control blocks, message information, etc.

Example:

```
ETB_STUBLOG=2
```

If the trace level is greater than 1, unencrypted contents of the send/receive buffers may be exposed in the trace.

Trace output, file *<thread-id>.etb*, is written to the trace directory. The location of the trace file depends on the settings of environment variable `%USERPROFILE%`, for example:

```
C:\Documents And Settings\<UserName>\My Documents\Software AG\EntireX
```

See *Trace Directory*.

Remember to switch off tracing to prevent trace files from filling up your disk.

To switch off tracing for the broker stub

- Set the environment variable `ETB_STUBLOG` to `NONE` or delete it.

Tracing Enterprise JavaBeans

To switch on tracing

- Set entry name `Trace` (see *Environment Entries to Control EJB*)
  - for minimal trace output to "1"
  - for detailed trace output to "2"
  - for full trace output to "3".

To switch off tracing

- Set entry name Trace to "0".

  Or:
  Omit the entry name Trace.

Trace Output

The trace output will be written to STDOUT.

To change the directory and name of the trace destination

- Set the entry name `Logfile` to a valid file name, depending on your operating system.

Logging Enterprise JavaBeans

To switch on logging

- Set entry name `Verbose` to true. (See *Environment Entries to Control EJB*.)

To switch off logging

- Set entry name `Verbose` to false.

  Or:
  Omit the entry name `Verbose`.

Log Output

The log output will be written to STDOUT.

Tracing EntireX Java ACI

The EntireX Java ACI provides a system property for tracing.

To switch on tracing

1. When starting the Java virtual machine, set the Java system property `entirex.trace`
   - for minimal trace output to "1"
   - for detailed trace output to "2"
   - for full trace output to "3".
2. The programming interface of the EntireX Java ACI allows you to set the trace value by the Java application using the EntireX Java ACI, see *Tracing* under *Writing Advanced Applications - EntireX Java ACI*. There may also be other methods to provide the trace value. See your application documentation.

To switch off tracing

- Set the Java system property `entirex.trace` to 0 when starting the Java virtual machine

  Or:
  Omit the Java system property `entirex.trace` when starting the Java virtual machine.

Trace Output

The trace output will be written to STDOUT.

Tracing Java RPC Server

To switch on tracing

- When starting the Java virtual machine, set the Java system property `entirex.trace`
  - for minimal trace output to "1"
  - for detailed trace output to "2"
  - for full trace output to "3".

  See *Customizing the Java RPC Server*.

To switch off tracing

- Set the Java system property `entirex.trace` to "0" when starting the Java virtual machine.

  Or:
  Omit the Java system property `entirex.trace` when starting the Java virtual machine.

Trace Output

The trace output will be written to STDOUT.

Tracing the RPC Runtime

To switch on tracing

- Before starting the client application, set the environment variable `ERX_TRACELEVEL` to
  - `STANDARD` for minimal trace output
  - `ADVANCED` for detailed trace output
  - `SUPPORT` for full trace output.

To switch off tracing

- Set the environment variable to NONE or delete it.

Trace Output

By default the trace file, *ERXTrace.nnn.log*, will be written to the trace directory.

The value *nnn* can be in the range from 001 to 005.

To change the trace destination

- Set the environment variable `ERX_TRACEFILE` to the desired destination, which can consist of file names, folder names and variables for file names, folder names, process ID, thread ID, range.

  The variables are:

Variable	Operating System	Description
%...%	Windows	environment variable
$(...)	UNIX	environment variable
@PID	UNIX, Windows	process ID
@TID	UNIX, Windows	thread ID
@RANGE[*n*,*m*]	UNIX, Windows	*m* must be greater than *n*, range is from 0 - 999
@CSIDL_PERSONAL	Windows	The user's home directory. The variable will be resolved by Windows shell functions.
@CSIDL_APPDATA	Windows	The *Application Data Directory*. The variable will be resolved by Windows shell functions.
@CSIDL_LOCAL_APPDATA	Windows	The *Application Data Directory*. The variable will be resolved by Windows shell functions.

Related Information

Environment Variables in EntireX

Tracing the RPC Server

To switch on tracing

- Set the `TraceLevel` parameter in the server configuration file to
  - `STANDARD` for minimal trace output
  - `ADVANCED` for detailed trace output
  - `SUPPORT` for full trace output.

  See *Configuring the RPC Server*.

Tracing can also be switched on and off with the environment variable *ERX_TRACELEVEL*. The settings in the configuration file override the environment variable.

To switch off tracing

- Set the `TraceLevel` parameter in the server configuration file to NONE.

Trace Output

By default the trace file, *ERXTrace.nnn.log*, will be written to the trace directory.

The value *nnn* can be in the range from 001 to 005.

To change the trace destination

- Set the parameter `TraceDestination` in the server configuration file to the desired destination. See *Configuring the RPC Server*.

  The variables are:

Variable	Operating System	Description
%...%	Windows	environment variable
$(...)	UNIX	environment variable
@PID	UNIX, Windows	process ID
@TID	UNIX, Windows	thread ID
@RANGE[*n*,*m*]	UNIX, Windows	*m* must be greater than *n*, range is from 0 - 999
@CSIDL_PERSONAL	Windows	The User's Home Directory. The variable will be resolved by Windows shell functions.
@CSIDL_APPDATA	Windows	The *Application Data Directory*. The variable will be resolved by Windows shell functions.
@CSIDL_LOCAL_APPDATA	Windows	The *Application Data Directory*. The variable will be resolved by Windows shell functions.

Related Information

EntireX RPC Server Return Codes

Tracing the XML/SOAP Runtime

This section provides information on tracing the following components:

- EntireX XML/SOAP RPC Server
- EntireX XML/SOAP Listener
- EntireX XML/SOAP Wrapper

The following topics are covered:

- Enabling Tracing
- Disabling Tracing
- Configuring a Trace File for the XML/SOAP Listener
- Configuring a Trace File for the XML/SOAP Wrapper or the XML/SOAP RPC Server
- Trace Parameters
- Component Names

Enabling Tracing

There are two ways to switch on tracing mode:

- Using a Property File
- Using Trace Parameters of the Java Virtual Machine

Using a Property File

To switch on tracing mode using a property file

1. Copy the trace property file *entirex.trace.standard* to one of the following locations:
   - the working directory of your client application;
   - the user's home directory;
   - any other location.
2. Rename the copied file *entirex.trace.properties*.
3. Customize *entirex.trace.properties* as described in *Trace Parameters*.
4. If *entirex.trace.properties* is within the current directory of your client application or your user home directory, it will be located automatically.

   Otherwise, specify the fully qualified or relative file name when starting the Java virtual machine for your client application using property `entirex.sdk.default.trace.propertiesfile`, example:

```
java -Dentirex.sdk.default.trace.propertiesfile ="D:/MyDirectory/entirex.trace.properties" MyClient
```

Using Trace Parameters of the Java Virtual Machine

To switch on tracing mode by specifying the trace parameters of the Java virtual machine

- Submit the trace parameters when you start the Java virtual machine for the application to be traced. See *Trace Parameters*. Note that parameter specifications submitted overwrite settings in the property file.

Disabling Tracing

To switch off tracing

- Delete or rename the trace property file if it is located in the working directory or in the user's home directory.

  Or:
  Specify `level=NONE` when invoking the Java virtual machine :

  ```
  java -Dentirex.sdk.default.trace.level = NONE MyClient
  ```

Configuring a Trace File for the XML/SOAP Listener

We recommend to add the following parameter in file *conf/axis2.xml* located in the Software AG Common Web Services Stack installation:

```
<parameter name="exx-trace-propertiesfile">file:///path of trace.properties file</parameter>
```

Example:

```
<parameter name="exx-trace-propertiesfile">file:///D:/MyDir/entirex.trace.properties</parameter>
```

Notes:

1. If a relative path is specified, the file is located in directory *WEB-INF/conf/* in the Web Services Stack web application file that contains the property.
2. In the parameter section of the file *axis2.xml*, the value of the parameter `exx-trace-propertiesfile` can contain definitions of operating system variables, for example `location="$EXXDIR/config/entirex.trace.properties"`.

Configuring a Trace File for the XML/SOAP Wrapper or the XML/SOAP RPC Server

See *Enabling Tracing*.

Note:
If the XML/SOAP RPC Server is running as a service, enable tracing by adding a Java system property to the start script or by copying file `entirex.trace.properties` to the same directory as the start script.

Trace Parameters

The following table provides an overview on trace parameters, their respective values, and how to submit them as arguments when invoking the Java virtual machine for the component to be traced.

Parameter	Syntax	Description
propertiesfile	entirex.sdk.*component name*.trace.propertiesfile= *absolute or relative path including the properties file*	Provide the location of the *entirex.trace.properties* file. Only used when the component is started. **Note:** A sample trace property file named *entirex.trace.standard* with predefined trace settings is contained in the directory *../EntireX/config*. This file is a model and must be renamed to the valid name when used.
level	entirex.sdk.*component name*.trace.level = *tracelevel*	You can specify the following trace levels: Keyword / Level / Description NONE / No tracing / Tracing is switch off STANDARD / User / Trace invocation of a component. ADVANCED / Expert / For support and diagnostics. Expert knowledge of the component is required. SUPPORT / Expert / Full trace output. Otherwise, as above.
directory	entirex.sdk.*component name*.trace.directory = *absolute or relative path*	Default is the working directory.

Parameter	Syntax	Description
filename	entirex.sdk.*component name*.trace.filename = *FILE|STDOUT|STDERR*	Specify where tracing data is written to: Keyword — Destination STDOUT (Default) — Console STDERR — Console FILE — File name is generated internally: *exx.sdk.component name.threadName. backupNo.log* , where `backupNo` is in the range from ".000" to ".009". Note that the number of files created depends on *`maximumsize`*. If more than 10 files are required, the oldest backup file is overwritten.
threadoriented	entirex.sdk.*component name*.trace.threadoriented = true | false	Keyword — Description YES — Thread-oriented: trace data is distributed over multiple files (one file per thread) NO (Default) — Trace data is written to one file.
rowlength	entirex.sdk.*component name*.trace.rowlength = *maximum_characters_per_row*	Maximum number of characters per row. If this limit is exceeded, the remaining letters are written to a new line.
maximumsize	entirex.sdk.*component name*.trace.maximumsize = *max_file_size*	Maximum size of the log file. If this limit is exceeded, the log file is renamed and the remaining data is written to a new log file, see *`filename`*. Note that this specification has an effect only if `filename` is set to "FILE".

Parameter	Syntax	Description
timeframe	entirex.sdk.*component name*.trace.timeframe = *number of day*	Time period after which the log file is closed. If this time limit has exceeded, the log file is renamed and the remaining data (if any) is written to a new log file. Note that this specification has an effect only if `filename` is set to "FILE". You can specify the following timeframes: Keyword Description 1-9+H Number of hours 1-9+D Number of days If no time frame is defined, only one log file is used until tracing is stopped. Example: If timeframe has been set to `30D`, the current log file is closed and renamed at midnight every thirty days, and tracing is continued with a new log file.

Component Names

Trace properties given in the trace property file might have to be restricted by *componentname*. The following components are available:

EntireX Component	componentname	Description
	default	The trace property is not restricted to a specific EntireX component.
XML/SOAP Runtime	xml.runtime	The trace property belongs to the EntireX XML/SOAP Runtime only.

Tracing the EntireX RPC-ACI Bridge

To trace Broker calls

1. Use the system property `entirex.trace=[0|1|2|3]`.

   This trace does not separate the calls to the Broker for RPC from those to the Broker for ACI. The trace levels are:

   - 0 to switch off tracing.
   - 1 to trace Broker calls.

- 2 to trace Broker calls and the payload.
- 3 to trace Broker calls and all buffers including the payload.

2. Redirect the trace to a file with the property `entirex.server.logfile`. Set this to the file name of the log file, the default is standard output.