webMethods EntireX

Monitoring with Command-line Scripts

Version 10.3

October 2018

WEBMETHODS

This document applies to webMethods EntireX Version 10.3 and all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Document ID: EXX-SCRIPTING-103-20191129

Table of Contents

1 About this Documentation

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
`Monospace font`	Identifies service names and locations in the format *`folder.subfolder.service`*, APIs, Java classes, methods, properties.
`Italic`	Identifies: Variables for which you must supply values specific to your own situation or environment. New terms the first time they occur in the text. References to other documentation sources.
`Monospace font`	Identifies: Text you must type in. Messages displayed by the system. Program code.
{ }	Indicates a set of choices from which you must choose one. Type only the information inside the curly braces. Do not type the { } symbols.
|	Separates two mutually exclusive choices in a syntax line. Type one of these choices. Do not type the | symbol.
[ ]	Indicates one or more options. Type only the information inside the square brackets. Do not type the [ ] symbols.
...	Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...).

Online Information and Support

Software AG Documentation Website

You can find documentation on the Software AG Documentation website at **http://documentation.softwareag.com**. The site requires credentials for Software AG's Product Support site Empower. If you do not have Empower credentials, you must use the TECHcommunity website.

Software AG Empower Product Support Website

If you do not yet have an account for Empower, send an email to empower@softwareag.com with your name, company, and company email address and request an account.

Once you have an account, you can open Support Incidents online via the eService section of Empower at **https://empower.softwareag.com/**.

You can find product information on the Software AG Empower Product Support website at **https://empower.softwareag.com**.

To submit feature/enhancement requests, get information about product availability, and download products, go to **Products**.

To get information about fixes and to read early warnings, technical papers, and knowledge base articles, go to the **Knowledge Center**.

If you have any questions, you can find a local or toll-free number for your country in our Global Support Contact Directory at **https://empower.softwareag.com/public_directory.asp** and give us a call.

Software AG TECHcommunity

You can find documentation and other technical information on the Software AG TECHcommunity website at **http://techcommunity.softwareag.com**. You can:

- Access product documentation, if you have TECHcommunity credentials. If you do not, you will need to register and specify "Documentation" as an area of interest.
- Access articles, code samples, demos, and tutorials.
- Use the online discussion forums, moderated by Software AG professionals, to ask questions, discuss best practices, and learn how other customers are using Software AG technology.
- Link to external websites that discuss open standards and web technology.

Data Protection

Software AG products provide functionality with respect to processing of personal data according to the EU General Data Protection Regulation (GDPR). Where applicable, appropriate steps are documented in the respective administration documentation.

2 Introduction

Scope

EntireX provides a set of command-line scripts as a solution to the following scenarios:

- "I want a quick overview of my standard broker and a list of active external services that are running."
- "I want to monitor an EntireX component (broker, service, client) over time."
- "I want to monitor my environment and check that all components (broker, RPC servers) are up and running."

You can select the scripts from the EntireX Command-line Scripts Menu or call the individual scripts from the command-line.

Note: You can use these scripts with local or remote brokers. The scripts were introduced with version 9.7, but can be used with brokers of any supported version.

EntireX Command-line Script Menu

To call the command-line script menu

- Choose **Start > Software AG > Administration > EntireX Command-line Scripts**.

  Or:

  Enter command `Menu_Of_EntireX_Command_Line_Scripts.bat` in the EntireX bin directory.

  The following screen appears:

```
List of EntireX Command-line Scripts


Current Broker:  localhost:1971

1 : Show Broker and registered Services

The following scripts write to CSV file:
2 : Monitor Broker
3 : Monitor Services
4 : Monitor Clients

5 : Change current Broker (this session only)
6 : Edit Broker and other defaults (persistent)

7 : Define your Environment
8 : Monitor your Environment

9 : Open new command window (in script directory)

0 : Exit

Enter the number to be executed or ? for help
```

The screen shows the current broker to which the administration tasks apply. You can change this for the duration of the session or persistently in the defaults. See *Using the Broker ID in Applications* in the EntireX Broker ACI Programming documentation for possible formats.

Enter a number to perform the respective task. The options are described in the table below, together with the corresponding script if you prefer to call the script directly.

Script Menu Options

Option	Description	Corresponding Script	Note
1	Show Broker and registered Services. Displays information on the current broker and its registered services.	`entirex_overview.bat`	
2	Monitor Broker. Monitors the current broker.	`monitor_broker_to_csv.bat`	Output for these three monitoring scripts is written to a CSV file (comma-separated values), which you can view, for example, with a spreadsheet tool.
3	Monitor Services. Monitors services registered to the current broker.	`monitor_service_to_csv.bat`	
4	Monitor Clients. Monitors clients registered to the current broker.	`monitor_client_to_csv.bat`	
5	Change Current Broker. Changes the broker used in the various scripts for this session only.		When the session is restarted, this value reverts to the default.
6	Edit Broker and other Defaults. You can modify defaults for Broker ID, timeout values and output files.	`edit_user_specific_monitor_defaults.bat`	
7	Define your Environment. You can define the environment (list of broker and registered services) you want to monitor.	`edit_user_specific_environment_definition.bat`	
8	Monitor your Environment (defined with option 7).	`monitor_environment.bat`	
9	Open New Command Window (in command-line script directory).		This enables you to start the scripts with your parameters.
0	Exit.		Exits the menu. If you changed the current broker with Option 5, this will revert to the default value. Monitoring scripts that write to CSV files running in a separate command window continue to run.

Changing the Current Broker

To change the current broker for your current session

- Choose option 5 from the *EntireX Command-line Script Menu*, "Change current Broker (this session only)". This changes the current broker ID used for executing the command-line scripts in this session.

  If you restart the script, the broker ID will be (re)set to the default value.

To change the current broker permanently

- Choose option 6 from the *EntireX Command-line Script Menu*, "Edit Broker and other Defaults (persistent)".

  Or:
  Run command `edit_user_specific_monitor_defaults.bat` in the EntireX bin directory.

  See *Default Handling*.

3 Show Broker and Registered Services

Scenario: "I want a quick overview of my standard broker and a list of active external services that are running."

With script `entirex_overview.bat`, EntireX offers a simple solution to show details of a specified or default broker and the active external services registered to it.

Calling the Script

To show a broker and its registered services

- Select option 1 from the *EntireX Command-line Script Menu*, "Show Broker and registered Services".

  Or:

  Enter one of the following commands:

```
entirex_overview.bat

entirex_overview.bat <BrokerId>

entirex_overview.bat <BrokerId> <UserId>

entirex_overview.bat <BrokerId> <UserId> <Password>
```

  where `<BrokerId>` is the ID of the broker to be monitored (default `localhost:1971`), and
  `<UserId>` is your user ID for broker calls if your broker is running with EntireX Security (no default)
  `<Password>` is your password (no default)

If no broker ID is specified, an overview of the default broker is provided. See *Default Handling* on how to change this.

See also *Using the Broker ID in Applications*.

Sample Output

Sample command:

```
entirex_overview.bat localhost:1971 myUserId myPassword
```

Sample output:

```
Overview of Broker localhost:1971

 Broker ID         : ETB001
 Running on        : pcusr1
 Version           : 10.3.0.00
 License expiration: UNLIMITED
 Trace level       : 0
 Platform          : PC       Windows 7 Professional
 Client timeout    : 900

 Transport Settings
 NET           : NO
 SSL           : NO
 TCP           : YES

 Dynamic Memory Management: YES
 Dynamic Worker Management: YES
 Re-read attribute file   : YES

 Attribute file: C:\SoftwareAG\EntireX\config\etb\ETB001\ETB001.atr
 Log file      : C:\SoftwareAG\EntireX\config\etb\ETB001\ETB001.log
 License file  : C:\SoftwareAG\common\conf\exx103.xml

 Workers       : Active: 1
 Services      : Active: 9

 High Watermarks
 Servers       : 2
 Clients       : 1
 Conversations : 1
 Memory        : 30153112

List of Active (External) Services for Broker localhost:1971


 Class         : RPC
 Server        : XMLSERVER
 Service       : CALLNAT

 No. of times all server instances were busy: 0
 Maximum of pending parallel conversations  : 0
```

```
 Total No. of requests                      : 0
 No. of active server instances             : 2
 Conversation high watermarks               : 0

End of overview
```

4 Monitoring EntireX Components

Scenario: “I want to monitor an EntireX component (broker, service, client) over time.”

EntireX provides multiple scripts to monitor - at a specified interval - your standard broker, registered services, and clients that call your broker. Output is written to a CSV file.

Monitoring Broker

Scenario: “I want to monitor my standard broker over time.”

Script `monitor_broker_to_csv.bat` writes key broker usage information to a CSV file. The report includes information such as active workers, clients, servers, allocated storage etc. The report is appended at a specified interval until the script is stopped.

Calling the Script

To monitor your current broker

- Select option 2 from the *EntireX Command-line Script Menu*, “Monitor Broker”.

  Or:

  Enter one of the following commands:

```
monitor_broker_to_csv.bat

monitor_broker_to_csv.bat <BrokerId>

monitor_broker_to_csv.bat <BrokerId> <Time> <UserId> <Password>
```

  where `<BrokerId>` is the ID of the broker to be monitored (default `localhost:1971`), and
  `<Time>` is the interval between reports in seconds (default 60)
  `<UserId>` is your user ID for broker calls if your broker is running with EntireX Security (no default)
  `<Password>` is your password (no default)

The first time you execute this script in a session, the results are displayed on screen so you can verify that the correct data is returned. You can override this behavior using environment variable `MONITOR_VERIFY`. Example:

```
set MONITOR_VERIFY=NO
```

The results of subsequent executions are written to a CSV file, with a new line created for each call. Default is *<drive>:\Users\user_id\documents\SoftwareAG\EntireX\out_monitor_broker.csv*. See also ***Default Handling***. Use environment variable `MONITOR_BROKER_OUTFILE` to specify a different output file. Example:

```
set MONITOR_BROKER_OUTFILE=c:\my_monitor_broker_outfile.csv
```

The content is based on broker information object `BROKER-OBJECT (Struct INFO_BKR)`.

CSV Column	**Field Name of** `BROKER-OBJECT`
Uptime (seconds)	`RUNTIME`
Active Workers	`NUM-WORKER-ACT`
Servers	`SERVER-ACT`
Server HWM	`SERVER-HIGH`
Clients	`CLIENT-ACT`
Client HWM	`CLIENT-HIGH`
Services	`SERVICE-ACT`
Conversation HWM	`CONV-HIGH`
Allocated Storage (bytes)	`TOTAL-STORAGE-ALLOCATED`
Storage HWM (bytes)	`TOTAL-STORAGE-ALLOCATED-HIGH`

where HWM=high watermark

The script will run until it is cancelled, for example with `ctrl+C` or by closing the command window.

Example

Sample command:

```
example monitor_broker_to_csv.bat localhost:1971 10 myUserId myPassword
```

Sample output:

	Time	Uptime	Active w	Servers	Server HWM	Clients	Client HWM	Services	Conversat	Allocated stor	Storage HWM
2	15:12.1	6098	1	2	17	1	31	9	59	29795200	334446456
3	16:13.2	6159	1	2	17	1	31	9	59	29795200	334446456
4	17:14.5	6220	1	32	32	61	61	12	62	368009320	368009320
5	18:16.2	6282	1	32	32	61	61	12	62	368009320	368009320
6	19:19.7	6345	1	32	32	54	61	12	62	359095928	368009320
7	20:22.3	6408	1	29	32	19	61	12	62	354639232	368009320
8	21:23.2	6469	1	3	32	7	61	10	62	307706280	368009320
9	22:24.2	6530	1	3	32	6	61	10	62	222753768	368009320
10	23:25.2	6591	1	3	32	6	61	10	62	90098240	368009320
11	24:26.2	6652	1	3	32	6	61	10	62	85641544	368009320
12	25:27.2	6713	1	3	32	6	61	10	62	85641544	368009320
13	26:28.2	6774	1	3	32	1	61	10	62	85641544	368009320
14											

Monitoring Services

Scenario: “I want to monitor the services registered to my standard broker over time.”

Script `monitor_service_to_csv.bat` writes key service usage information on external services registered to the current broker to a CSV file. The report includes information such as Class/Server/Service, active servers, number of requests, number of times the server was busy, pending conversations etc. The report is appended at a specified interval until the script is stopped.

By default, services with `CLASS=RPC` and `SERVICE=CALLNAT` are monitored.

Calling the Script

To monitor the services registered your current broker

- Select option 3 from the *EntireX Command-line Script Menu*, “Monitor Services”.

  Or:

  Enter one of the following commands:

```
monitor_service_to_csv_file.bat
```

```
monitor_service_to_csv_file.bat <BrokerID>
```

```
monitor_service_to_csv_file.bat <BrokerID> <Time> <Class> <Server> <Service> ↩
<UserId> <Password>
```

  where `<BrokerId>` is the ID of the broker to be monitored (default `localhost:1971`), and

`<Time>`	is the interval between reports in seconds (default 60)
`<Class>`	is the class to be monitored (default RPC)
`<Server>`	is the server to be monitored (default *)
`<Service>`	is the service to be monitored (default CALLNAT)
`<UserId>`	is your user ID for broker calls if your broker is running with EntireX Security (no default)
`<Password>`	is your password (no default)

The first time you execute this script in a session, the results are displayed on screen so you can verify that the correct data is returned. You can override this behavior using environment variable `MONITOR_VERIFY`. Example:

```
set MONITOR_VERIFY=NO
```

The results of subsequent executions are written to a CSV file, with a new line created for each (active) Service. Default is *<drive>:\Users\user_id\documents\SoftwareAG\EntireX\out_monitor_service.csv*. Specify a different output file with environment variable `MONITOR_SERVICE_OUTFILE`. Example:

```
set MONITOR_SERVICE_OUTFILE=c:\my_monitor_service_outfile.csv
```

The content is based on broker information object `SERVICE-OBJECT (Struct INFO_SV)`.

CSV Column	**Field Name of** `SERVICE-OBJECT`
Class	`SERVER-CLASS`
Server	`SERVER-NAME`
Service	`SERVICE`
Active Servers	`SERVER-ACT`
Server Busy (count)	`NUM-SERV-OCC`
Requests	`REQ-SUM`
Pending Conversations	`NUM-PEND`
Pending Conversations HWM	`PEND-HIGH`
Active Conversations	`CONV-ACT`
Conversation HWM	`CONV-HIGH`
Server Wait (count)	`NUM-WAIT-SERVER`

where HWM=high watermark

The script will run until it is cancelled, for example with `ctrl+C` or by closing the command window.

Example

Sample command:

```
monitor_service_to_csv_file.bat localhost:1971 10 RPC * * MyUser MyPassword
```

Sample output:

1	Time	Class	Server	Service	Active Ser	Server Bus	Requests	Pending C	Pending C	Active Cor	Conversat	Server Wait
2	15:14.2	RPC	XMLSERVE	CALLNAT	2	0	0	0	0	0	0	0
3	16:15.1	RPC	XMLSERVE	CALLNAT	2	0	0	0	0	0	0	0
4	17:16.7	RPC	XMLSERVE	CALLNAT	2	0	0	0	0	0	0	0
5	17:16.8	ACLASS	ASERVER	ASERVICE	10	13	85	0	10	8	22	85
6	17:16.8	CCLASS	CSERVER	CSERVICE	10	11	73	0	9	3	20	73
7	17:16.8	BCLASS	BSERVER	BSERVICE	10	27	90	0	8	7	20	90
8	18:20.0	RPC	XMLSERVE	CALLNAT	2	0	0	0	0	0	0	0
9	18:20.3	ACLASS	ASERVER	ASERVICE	10	21	589	0	10	5	22	589
10	18:20.3	CCLASS	CSERVER	CSERVICE	10	45	669	0	9	6	21	669
11	18:20.5	BCLASS	BSERVER	BSERVICE	10	49	598	0	8	7	20	598
12	19:24.2	RPC	XMLSERVE	CALLNAT	2	0	0	0	0	0	0	0
13	19:24.4	ACLASS	ASERVER	ASERVICE	10	86	1130	0	10	6	22	1130
14	19:24.4	CCLASS	CSERVER	CSERVICE	10	238	1328	0	9	8	21	1328
15	19:24.7	BCLASS	BSERVER	BSERVICE	10	102	1094	0	8	5	20	1094
16	20:27.3	RPC	XMLSERVE	CALLNAT	2	0	0	0	0	0	0	0
17	20:27.4	ACLASS	ASERVER	ASERVICE	10	86	1792	0	10	8	22	1792
18	20:27.4	CCLASS	CSERVER	CSERVICE	8	240	1873	0	9	7	21	1873
19	20:27.5	BCLASS	BSERVER	BSERVICE	9	102	1584	0	8	9	20	1584
20	21:29.2	RPC	XMLSERVE	CALLNAT	2	0	0	0	0	0	0	0
21	21:29.2	BCLASS	BSERVER	BSERVICE	1	110	1807	5	8	6	20	1807
22	22:30.2	RPC	XMLSERVE	CALLNAT	2	0	0	0	0	0	0	0
23	22:30.2	BCLASS	BSERVER	BSERVICE	1	110	1807	5	8	5	20	1807
24	23:31.2	RPC	XMLSERVE	CALLNAT	2	0	0	0	0	0	0	0
25	23:31.2	BCLASS	BSERVER	BSERVICE	1	110	1807	5	8	5	20	1807
26	24:32.3	RPC	XMLSERVE	CALLNAT	2	0	0	0	0	0	0	0
27	24:32.3	BCLASS	BSERVER	BSERVICE	1	110	1807	5	8	5	20	1807
28	25:33.2	RPC	XMLSERVE	CALLNAT	2	0	0	0	0	0	0	0

Monitoring Clients

Scenario: “I want to monitor the clients calling my standard broker over time.”

Script `monitor_client_to_csv.bat` writes key usage information on clients calling the current broker at a defined interval to a CSV file. The report includes information such as user ID, token, wait time, Class/Server/Service, hostname, environment information, start time and IP address etc. The report is appended at a specified interval until the script is stopped.

Calling the Script

To monitor the clients calling your current broker

- Select option 4 from the *EntireX Command-line Script Menu*, “Monitor Clients”.

  Or:

  Enter one of the following commands:

```
monitor_client_to_csv.bat
```

```
monitor_client_to_csv.bat <BrokerId>
```

```
monitor_client_to_csv.bat <BrokerId> <Time> <UserId> <Password>
```

  where `<BrokerId>` is the ID of the broker to be monitored (default `localhost:1971`), and
  `<Time>` is the interval between reports in seconds (default 60)
  `<UserId>` is your user ID for broker calls if your broker is running with EntireX Security (no default)
  `<Password>` is your password (no default)

The first time you execute this script in a session, the results are displayed on screen so you can verify that the correct data is returned. You can override this behavior using environment variable `MONITOR_VERIFY`. Example:

```
set MONITOR_VERIFY=NO
```

The results of subsequent executions are written to a CSV file, with a new line created for each (active) client. Default is *<drive>:\Users\user_id\documents\SoftwareAG\EntireX\out_monitor_service.csv*. See also ***Default Handling***. Use environment variable `MONITOR_CLIENT_OUTFILE` to specify a different output file. Example:

```
set MONITOR_CLIENT_OUTFILE=c:\my_monitor_client_outfile.csv
```

The content is based on broker information object `CLIENT-SERVER-PARTICIPANT-OBJECT (Struct INFO_CS)`.

CSV Column	Field Name of `BROKER-OBJECT`
UserID	`USER-ID`
Token	`TOKEN`
Unique User ID	`P-USER-ID`
Status	`STATUS`
Wait Conversation Type	`WAIT-CONV-TYPE`
Wait Class	`WAIT-SERVER-CLASS`
Wait Server	`WAIT-SERVER-NAME`
Wait Service	`WAIT-SERVICE`
Last Active (seconds)	`LAST-ACTIVE`
Sum Conversations	`SUM-CONV`
HostName	`HOST-NAME`
Application	`APPLICATION-NAME`
Application Type	`APPLICATION-TYPE`
Application Version	`APPLICATION-VERSION`
Start Time	`CREATE-TIME`
IPV4	`IP-ADDRESS`
IPV6	`IPV6-ADDRESS`

The script will run until it is cancelled, for example with `ctrl+C` or by closing the command window.

Example

Sample command:

```
example monitor_client_to_csv.bat localhost:1971 10 myUserId myPassword
```

Sample output (truncated):

1	Time	UserID	Token	Unique user ID	Status	Wait conv	Wait Class	Wait Serve	Wait Servi	Last activ	Sum conv	Host name	Application	Applicatio
2	15:18.1	EUR\usr		206D6370686F3	0					0	1	mcusr01	etbinfo.exe	WIN64
3	16:19.2	EUR\usr		206D6370686F3	0					0	1	mcusr01	etbinfo.exe	WIN64
4	17:20.5	STDCLT		206D6370686F3	0					6	2	mcusr01	nconvClt.exe	WIN64
5	17:20.5	STDCLT		206D6370686F3	0					7	2	mcusr01	nconvClt.exe	WIN64
6	17:20.6	STDCLT		206D6370686F3	0					9	1	mcusr01	nconvClt.exe	WIN64
7	17:20.6	STDCLT		206D6370686F3	0					8	1	mcusr01	nconvClt.exe	WIN64
8	17:20.7	STDCLT		206D6370686F3	5	NONE				7	9	mcusr01	nconvClt.exe	WIN64
9	17:20.7	STDCLT		206D6370686F3	0					7	1	mcusr01	nconvClt.exe	WIN64
10	17:20.8	STDCLT		206D6370686F3	0					9	1	mcusr01	nconvClt.exe	WIN64
11	17:20.8	STDCLT		206D6370686F3	0					11	1	mcusr01	nconvClt.exe	WIN64
12	17:20.8	STDCLT		206D6370686F3	0					11	1	mcusr01	nconvClt.exe	WIN64
13	17:20.8	STDCLT		206D6370686F3	0					11	1	mcusr01	nconvClt.exe	WIN64
14	17:20.9	STDCLT		206D6370686F3	0					11	1	mcusr01	nconvClt.exe	WIN64
15	17:20.9	STDCLT		206D6370686F3	5	NONE				0	9	mcusr01	nconvClt.exe	WIN64
16	17:20.9	STDCLT		206D6370686F3	0					11	1	mcusr01	nconvClt.exe	WIN64
17	17:20.9	STDCLT		206D6370686F3	0					11	1	mcusr01	nconvClt.exe	WIN64
18	17:21.0	STDCLT		206D6370686F3	5	NONE				0	10	mcusr01	nconvClt.exe	WIN64
19	17:21.0	STDCLT		206D6370686F3	0					0	11	mcusr01	nconvClt.exe	WIN64
20	17:21.1	STDCLT		206D6370686F3	5	NONE				1	10	mcusr01	nconvClt.exe	WIN64
21	17:21.1	STDCLT		206D6370686F3	0					0	7	mcusr01	nconvClt.exe	WIN64
22	17:21.2	STDCLT		206D6370686F3	0					0	3	mcusr01	nconvClt.exe	WIN64
23	17:21.2	STDCLT		206D6370686F3	0					2	2	mcusr01	nconvClt.exe	WIN64
24	17:21.2	STDCLT		206D6370686F3	5	NONE				0	12	mcusr01	nconvClt.exe	WIN64
25	17:21.2	STDCLT		206D6370686F3	0					0	18	mcusr01	nconvClt.exe	WIN64
26	17:21.3	STDCLT		206D6370686F3	0					11	1	mcusr01	nconvClt.exe	WIN64
27	17:21.3	STDCLT		206D6370686F3	0					1	6	mcusr01	nconvClt.exe	WIN64
28	17:21.4	STDCLT		206D6370686F3	5	NONE				0	20	mcusr01	nconvClt.exe	WIN64
29	17:21.4	STDCLT		206D6370686F3	0					0	6	mcusr01	nconvClt.exe	WIN64

Default Handling

You can customize the defaults used for the monitoring scripts.

To customize the defaults

1 Select option 6 in the ***EntireX Command-line Script Menu***, "Edit Broker and other defaults (persistent)".

Or:

Enter command `edit_user_specific_monitor_defaults`.

This copies file *default_values_for_monitor_to_csv_file.bat* to directory *<drive>:\Users\user_id\documents\SoftwareAG\EntireX* (if it does not already exist) and opens a text editor, for example Notepad.

2 Edit the file to match your environment settings. You can change defaults such as:

- broker ID
- default timeouts for the monitoring scripts
- output files for the monitoring scripts

The changes you make here are persistent: they apply to all subsequent sessions.

5 Monitoring your Environment

Scenario: "I want to monitor my environment and check that all components (broker, RPC servers) are up and running."

EntireX offers a script-based solution to check if all brokers and services of a defined environment are active.

Defining your Environment

To define the environment to be monitored

- Select option 7 from the *EntireX Command-line Script Menu*, "Define your Environment".

  Or:

  Enter command `edit_user_specific_environment_definition.bat` to specify the environment to be monitored (defined by broker and list of services).

  This opens a text editor (for example Notepad) with a sample definition of an environment that you can customize. You can enter values for the following parameters:

Parameter	Value	Description	Note
`ENVIRONMENT`	`<Env_Name>`	Logical name of the environment	
`LOGEXIT`	`<Exit_Name>`	User exit called after each component check.	Optional. See *Logging Exit*.
`ERROREXIT`	`<Exit_Name>`	Batch file to be called if a component of the environment is not active.	See *Error Handling*.
`BROKER`	`<Broker_Name> <Broker_ID>`	Logical name and ID of broker used for the etbinfo calls.	
	`<Broker_Name> <Broker_ID> <UserId> <Password>`	Additional user ID and password if the broker is running with EntireX Security.	
`SERVICE`	`<Service_Name> <Class> <Server> <Service>`	Logical service name, class, server, service to be monitored.	Checks if the specified service is registered at the broker.
`RPCSERVICE`	`<RPC_Service_Name> <Class> <Server> <Service>`	Logical RPC service name, class, server, service to be monitored.	Valid only for RPC servers and issues an RPC `ping` command to the specified service.

Notes:

1. The file may contain a list of environments.

2. Each environment can consist of list of brokers, and for each broker a list of services can be defined.
3. Blanks in the logical names are not supported.

The file you define here is used for the following scripts:

`monitor_environment.bat`	See *Monitoring your Environment*.
`process_environment_file.bat`	This batch file processes the environment definition file and calls `check_environment.bat`. This batch file is called by `monitor_environment.bat`.
`check_environment.bat`	This batch file is called by `process_environment_file.bat` with the parameters of one line of the environment definition file. The batch file checks the parameters and either: ■ sets environment variables for subsequent calls ■ calls etbinfo to check if the broker/service is running

Examples

This environment has one broker:

```
ENVIRONMENT myProductionServers
ERROREXIT handle_error.bat
BROKER myProductionBroker localhost:1971
RPCSERVICE myRPCServer RPC SRV1 CALLNAT
```

This environment has multiple brokers:

```
ENVIRONMENT myMFServers
ERROREXIT handle_error.bat
BROKER myMFBroker ibm2:3930
SERVICE myACIServer ACLASS ASERVER ASERVICE
BROKER myMFBroker2 ibm2:3940
SERVICE myACIServer2 ACLASS ASERVER ASERVICE
RPCSERVICE myRPCServer2 RPC SRV2 CALLNAT
```

Monitoring your Environment

To monitor your environment

- Select option 8 from the *EntireX Command-line Script Menu* “Monitor your Environment”.

  Or:

  Enter a command as shown below:

```
monitor_environment.bat
```

```
monitor_environment.bat <Time>
```

```
monitor_environment.bat <Time> <EnvDefFile>
```

  where `<Time>` is the interval between checks in seconds (default 60)

  `<EnvDefFile>` is the file containing the definition of the environment (default *MyEnvironment.cfg*).

  Example:

```
monitor_environment.bat 30 myEnvironmentDefinitionFile.txt
```

The following checks are performed:

- That the service is registered at the broker.
- That the server can be called. This is done with an RPC ping command.

A user exit specified in the environment definition file (see *Defining your Environment*) is called if a specified broker or service is not active. See *Error Handling* below.

Error Handling

A sample batch file `handle_error.bat` is provided to handle the situation where a component of a defined environment (see *Defining your Environment*) is not available. The environment definition file specifies the name of the error exit to be called. You can use this file as a template for your own exit to customize your error handling. We strongly recommend you rename this file.

```
@echo off
@rem the following environment variables are set when the bat file is called
@rem environment variable %OBJECT% Error Object. possible values: BROKER or SERVICE
@rem envirnoment variable ETBINFOERROR Error Number returned by ETBINO
@rem envrionment variable ETBINFOERRORTEXT Error text
@rem the following environment variables are set for OBJECT SERVICE and OBJECT BROKER
@rem environment variable %ENV% logical name of environment
@rem environment variable %BNAME% logical name of Broker
@rem environment variable %BID% Broker ID
@rem the following environment variables are only set for OBJECT SERVICE
@rem environment variable %SNAME% logical service name
@rem environment variable %CLASS% Class
@rem environment variable %SERVER% Server
@rem environment variable %SERVICE% Service

echo Example User exit to handle errors: handle_error.bat
echo Error during check of Environment %ENV%
echo Broker %BNAME% (%BID%)

@rem check error object
@rem %OBJECT% == BROKER - Error Situation: defined Broker cannot be called
if %OBJECT%.==BROKER. goto Broker
@rem %OBJECT% == SERVICE - Error Situation: defined Service not registered
if %OBJECT%.==SERVICE. goto Service
echo Unknown Error Object %OBJECT%
goto end

:Broker
@rem the Broker (logical Name BNAME, Broker ID BID) is not running.
@rem add your code here to handle this situation

echo FATAL ERROR
echo Environment %ENV%
echo Broker %BNAME% ( %BID%) not active
goto end

:Service
@rem the Service (logical Name SNAME ,  CLASS /  SERVER / SERVICE ) on
@rem Broker (logical Name BNAME, Broker ID BID) is not running.
@rem add your code here to handle this situation

echo FATAL ERROR
echo Environment %ENV%
echo Service %SNAME% (%CLASS% / %SERVER% / %SERVICE% ) at Broker %BNAME% ( %BID%) ↩
 not registered
goto end


:end
@rem remove the pause so that monitoring of the environment can continue without a ↩
break
pause
```

Logging Exit

You optionally specify a logging exit when you are monitoring your environment. The exit is called every time a component in the environment is checked (`BROKER`, `RPCSERVICE`, `SERVICE`). With the exit you can, for example, write a CSV file with the result of the checks. A sample script `log_environment.bat` is provided, which you can use as a template. We strongly recommend you rename this file.

The exit contains the following environment variables:

```
@rem the following environment variables are set when the bat file is called
@rem environment variable %OBJECT% Object. Possible values: BROKER or SERVICE
@rem environment variable %CHECK_ERROR% Error Flag. Possible values: TRUE or FALSE
@rem in case of error the following envrionment variable provide details about the ↩
Error.
@rem envirnoment variable ETBINFOERROR Error Number returned by ETBINO
@rem envrionment variable ETBINFOERRORTEXT Error text
@rem the following environment variables are set for OBJECT SERVICE and OBJECT BROKER
@rem environment variable %ENV% logical name of environment
@rem environment variable %BNAME% logical name of Broker
@rem environment variable %BID% Broker ID
@rem the following environment variables are only set for OBJECT SERVICE
@rem environment variable %SNAME% logical service name
@rem environment variable %CLASS% Class
@rem environment variable %SERVER% Server
@rem environment variable %SERVICE% Service
```