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. This chapter covers the following topics:

- Monitoring Broker
- Monitoring Services
- Monitoring Clients
- Default Handling

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
- Example

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

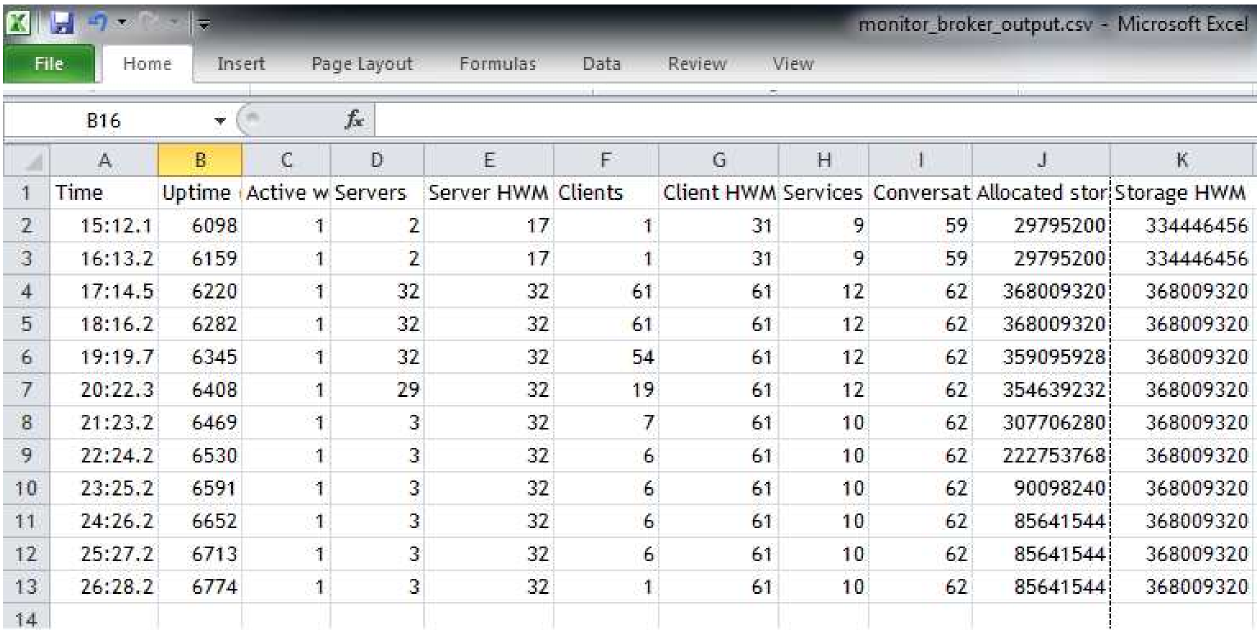
The script will run until it is cancelled, for example with `ctrl+C` or by closing the command window.

Example

Sample command:

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

Sample output:



	A	B	C	D	E	F	G	H	I	J	K
1	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
- Example

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

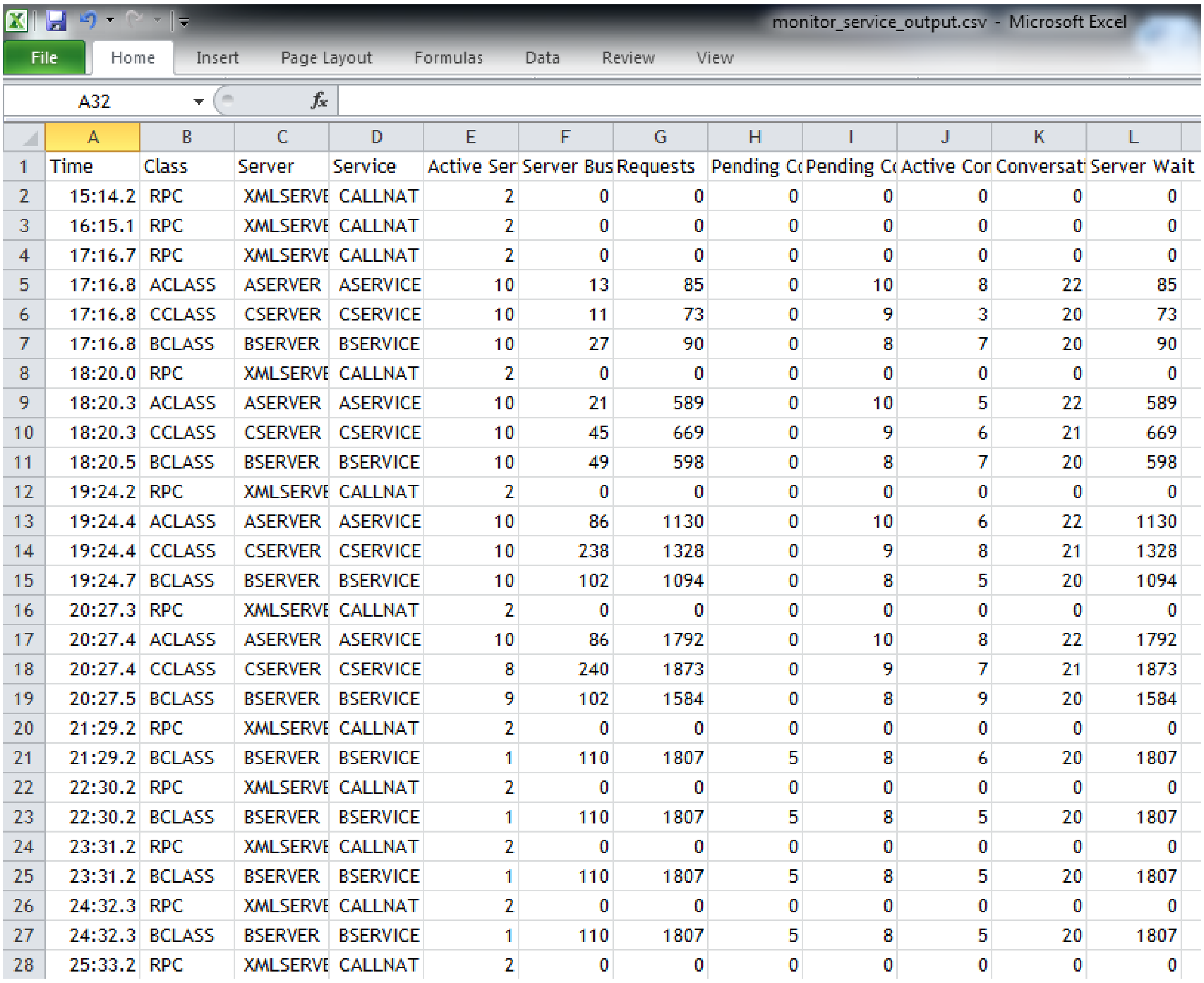
The script will run until it is cancelled, for example with `ctrl+C` or by closing the command window.

Example

Sample command:

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

Sample output:



	A	B	C	D	E	F	G	H	I	J	K	L
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
- Example

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

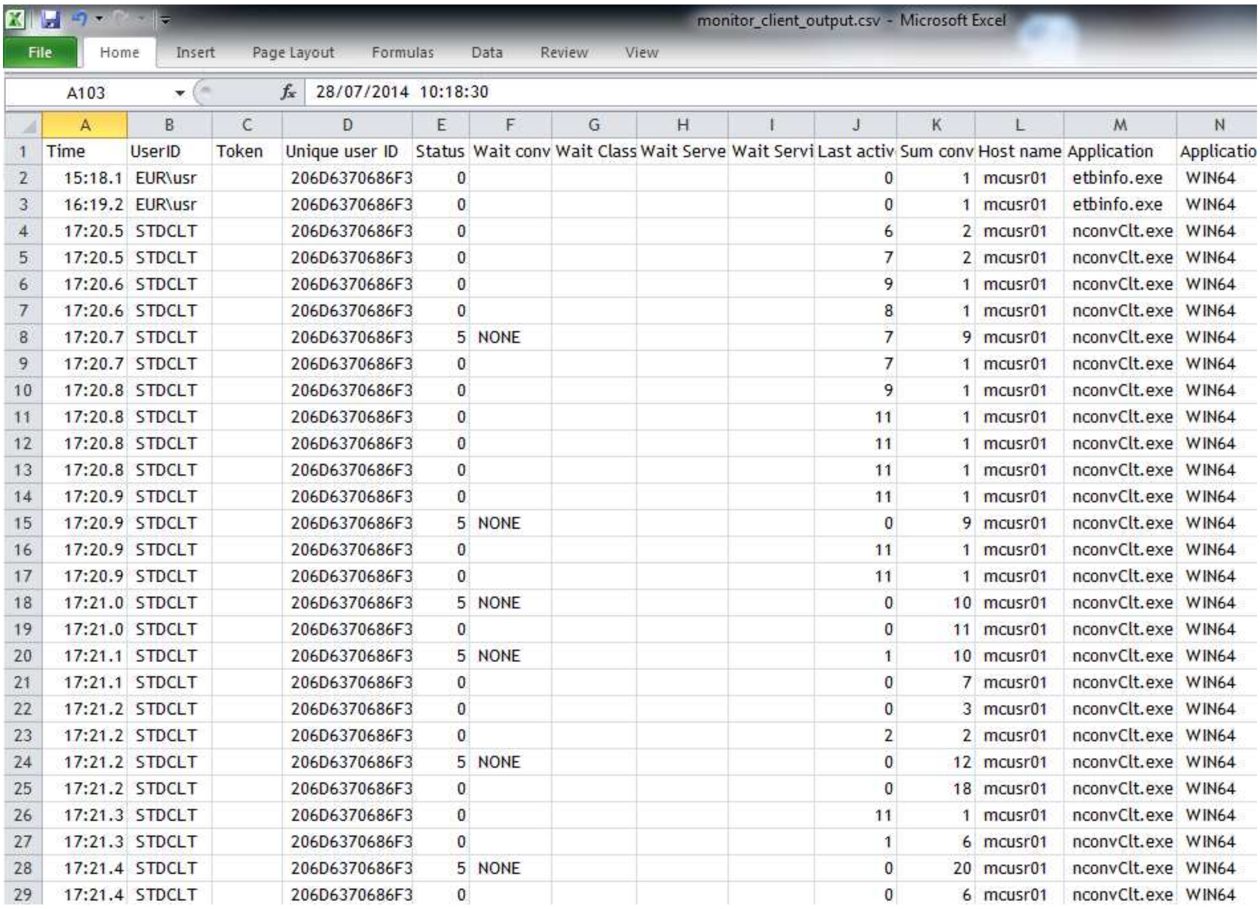
The script will run until it is cancelled, for example with `ctrl+C` or by closing the command window.

Example

Sample command:

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

Sample output (truncated):



A103 28/07/2014 10:18:30

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
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.