webMethods EntireX

Application Monitoring

Version 10.1

October 2017

WEBMETHODS

This document applies to webMethods EntireX Version 10.1 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-APPMON-101-20191129

Table of Contents

Preface

This documentation explains how to receive response-time data from your distributed applications. It is organized under the following headings:

Introduction	What is application monitoring? Sample scenarios in which the EntireX Broker and the EntireX Adapter are used. An overview of the response time KPIs.
Components that Support Application Monitoring	List of components that can be used to monitor distributed application scenarios.
Setting Up Application Monitoring	How to set up EntireX Broker and the EntireX Adapter. Information on the configuration file for the Application Monitoring Data Collector. How to start and stop the Application Monitoring Data Collector.
KPI Definitions for Application Monitoring	Describes the key performance indicators (KPIs) monitored by the Application Monitoring Data Collector.

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

What is Application Monitoring?

Application monitoring adds new value to monitoring. It enables you to monitor the response times in your distributed applications, and it also enables you to monitor a couple of error situations.

The EntireX Application Monitoring Data Collector collects the response time data of each involved software component of selected synchronous EntireX RPC services. The Application Monitoring Data Collector stores the KPI (key performance indicator) values in CSV (comma-separated values) files. The files can be processed by any tool which supports CSV files. A sample MashApp is provided in Empower (including documentation in PDF) under **Products & Documentation** > **Download Components** > **EntireX - Application Monitoring MashApp**.

When a service has been selected for monitoring, each call to the service by a client application is monitored. The overall service response times, the network transport times, the EntireX Broker processing and waiting times, the RPC (remote procedure call) server processing times, and the time spent for database calls are measured. Each involved Software AG enterprise product concatenates the monitored time(s) with the service call. When the call returns to the client, the client RPC runtime provides the event data to the Application Monitoring Data Collector.

In addition to monitoring RPC scenarios as described above, you can also monitor scenarios where the EntireX Adapter is used to call transactions using CICS ECI or IMS Connect.

The collection of response times and other measuring data from your distributed application in real time requires that you set up various components for application monitoring. See *Setting Up Application Monitoring* for detailed information.

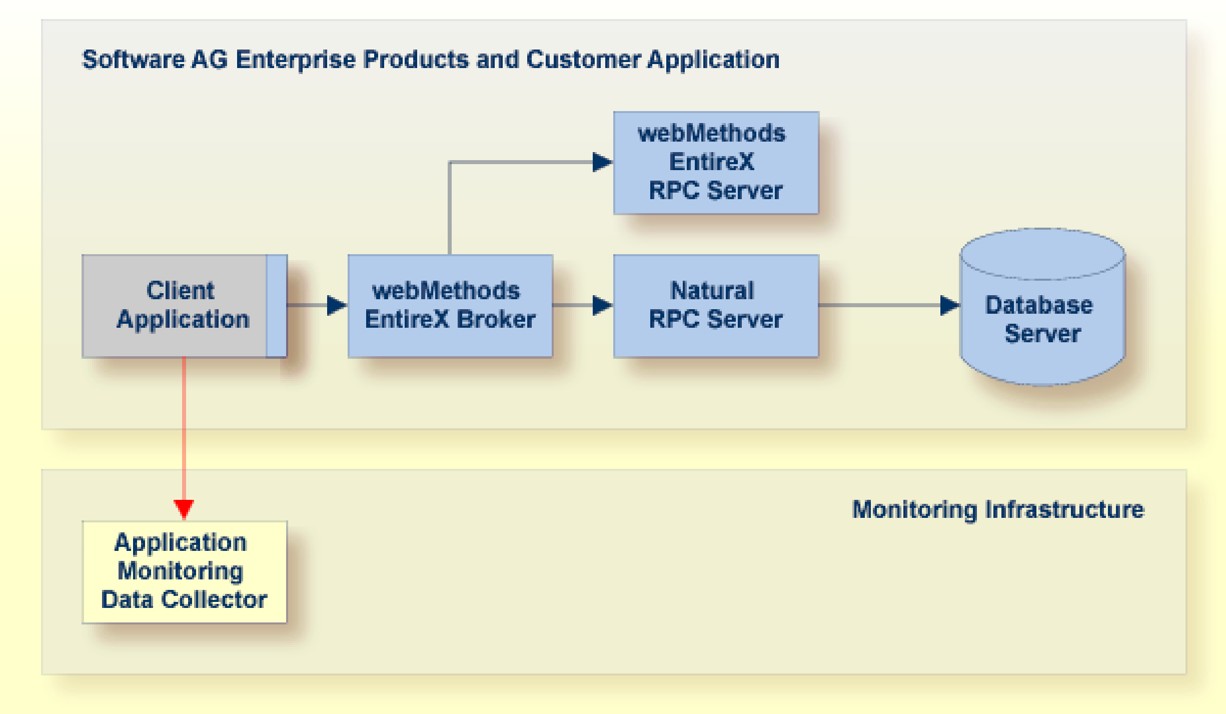
Sample Scenarios

The following graphics illustrate examples of different scenarios in which application monitoring can be used. The boxes in the graphics contain the components that are running in your production environment. The components in the blue boxes can be monitored. The components in the gray boxes cannot be monitored. The yellow box represents the Application Monitoring Data Collector which is required for collecting data and for measuring the response times. The black lines stand for calls from the application. The red line stands for the measuring data which are being transported.

Note: See also *Components that Support Application Monitoring*.

RPC Scenario with EntireX Broker

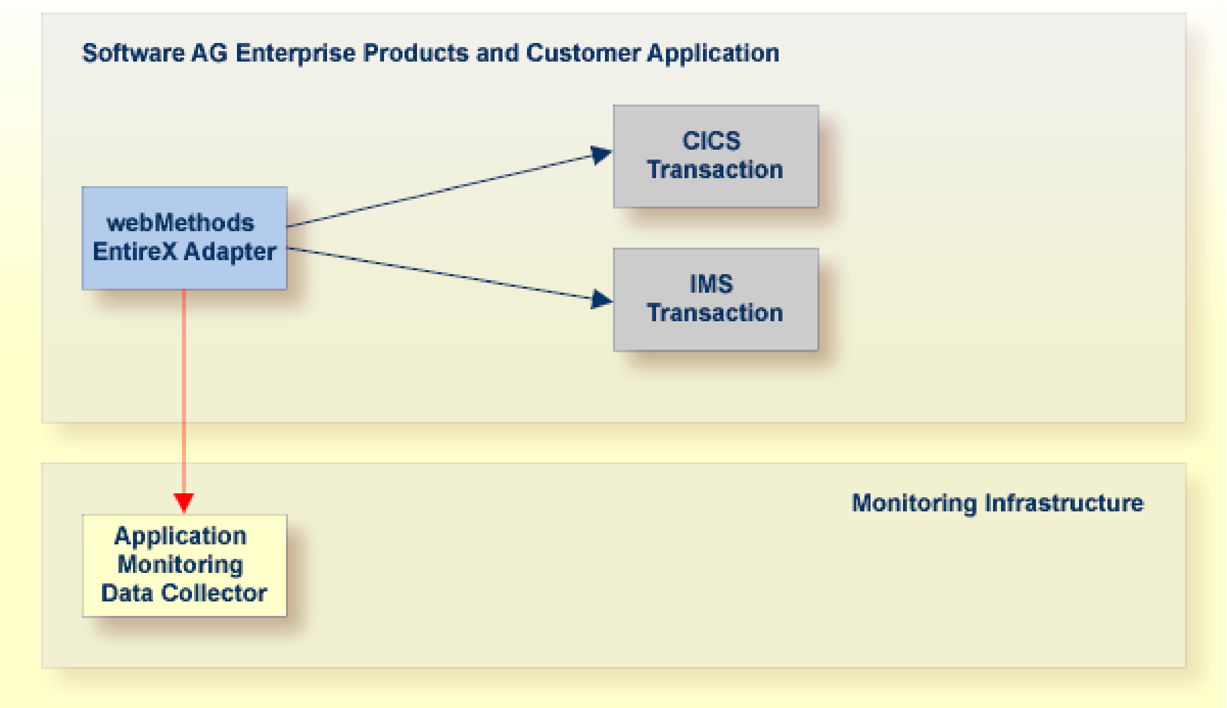
The small blue box linked to the gray "Client Application" box represents the client RPC layer which can be monitored, whereas the client application itself cannot be monitored. Note that the time spent for database calls can only be monitored if the call is issued by a Natural RPC server.


Software AG Enterprise Products and Customer Application
webMethods EntireX RPC Server
Client Application
webMethods EntireX Broker
Natural RPC Server
Database Server
Monitoring Infrastructure
Application Monitoring Data Collector

Note: In the above scenario, the Direct RPC component of the EntireX Adapter can be used instead of the EntireX Broker.

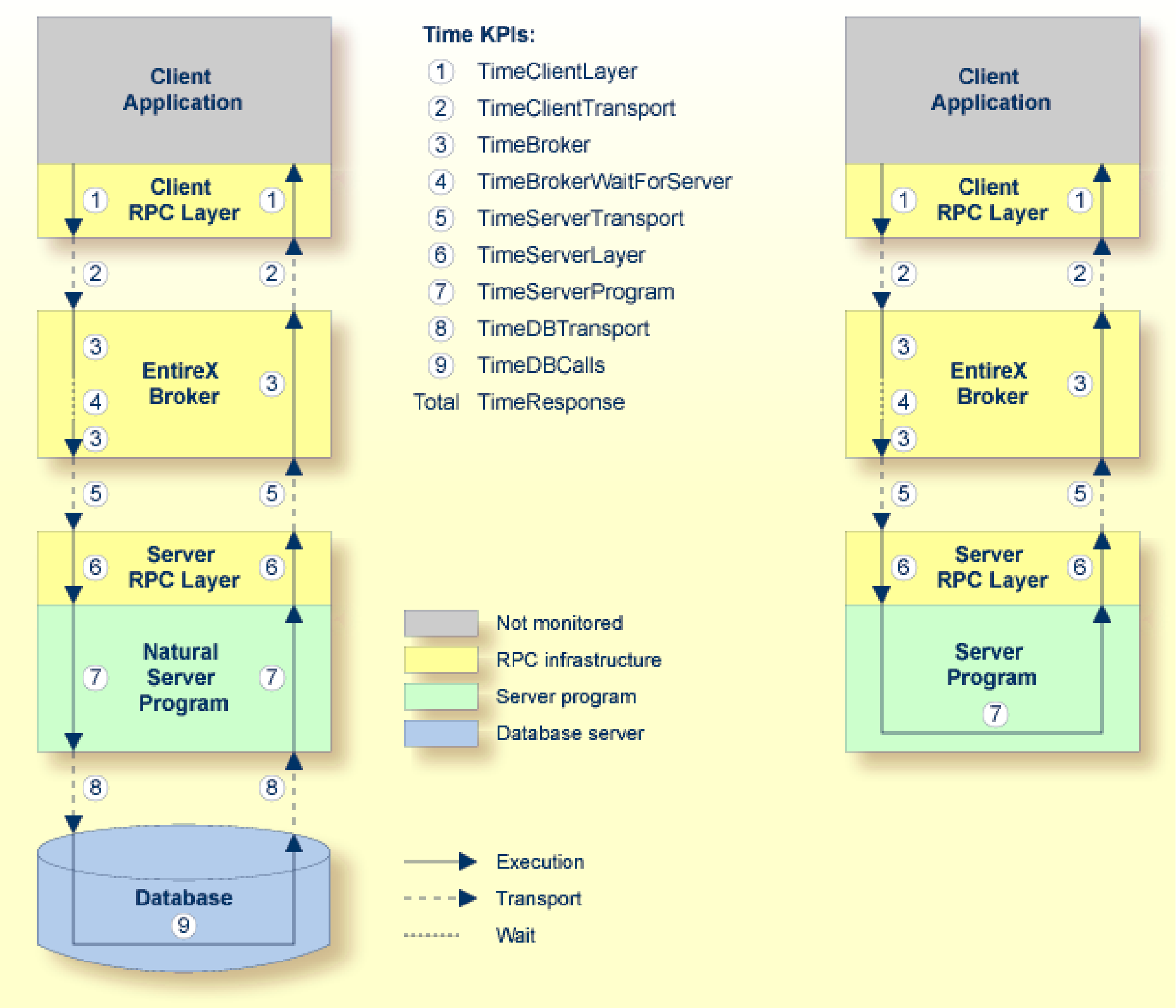
CICS ECI and IMS Connect Scenario with EntireX Adapter

The time spent in the CICS or IMS transaction cannot be monitored. Only the sum of the transport time to CICS ECI or IMS Connect plus the time spent in the CICS or IMS transaction can be monitored.


Software AG Enterprise Products and Customer Application
webMethods EntireX Adapter
CICS Transaction
IMS Transaction
Monitoring Infrastructure
Application Monitoring Data Collector

Response Time KPIs

The following graphic shows the detailed meanings of the response time KPIs that are collected by the Application Monitoring Data Collector. As an example, a remote procedure call (RPC) from a client application to a server program is illustrated. For a Natural server program, database calls can be monitored; this is illustrated on the left side of the graphic. Note that the database transport time (**TimeDBTransport** KPI) is only available for a database call against an Adabas server. For other databases, the database transport time is included in the database calls time (**TimeDBCalls** KPI). The right side illustrates a non-Natural server program where database calls cannot be monitored.

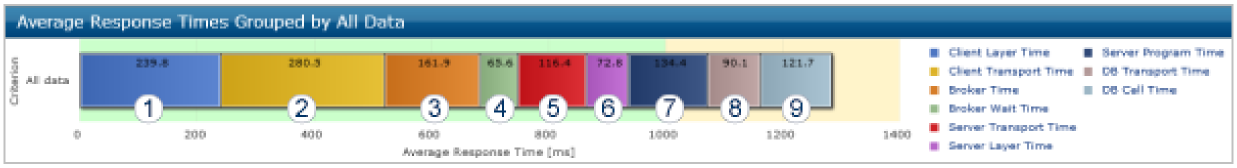


1. The client application issues a remote procedure call and the client RPC layer gets control. At the end of a transaction, the client RPC layer gives the control back to the client application. The time spent in the client RPC layer is monitored by the **TimeClientLayer** KPI.
2. The client RPC layer calls the EntireX Broker. The transport time between the client RPC layer and the EntireX Broker plus the transport time spent on the way back is monitored by the **TimeClientTransport** KPI.
3. The time spent in the EntireX Broker is monitored by the **TimeBroker** KPI. The KPI value does not include the amount of time that the EntireX Broker spends waiting for an available server.
4. The amount of time that the EntireX Broker spends waiting for an available server is monitored by the **TimeBrokerWaitForServer** KPI.
5. When the EntireX Broker calls the server program, the server RPC layer receives the call first. The transport time between the EntireX Broker and the server RPC layer plus the transport time spent on the way back is monitored by the **TimeServerTransport** KPI.

6. The time spent in the server RPC layer is monitored by the **TimeServerLayer** KPI.
7. The server RPC layer forwards control to the Natural server program. The time spent in the Natural server program is monitored by the **TimeServerProgram** KPI. The KPI value does not include the time spent for database calls.
8. The Natural server program calls a database. The transport time between the Natural server program and the database plus the transport time spent on the way back is monitored by the **TimeDBTransport** KPI. This KPI is only available for Natural RPC servers issuing database calls against an Adabas server.
9. The time spent for database calls is monitored by the **TimeDBCalls** KPI. For non-Adabas databases, the KPI value includes also the transport time required to reach the database server. This KPI is only available for Natural RPC servers.

- The **TimeResponse** KPI reflects the complete response time on the round trip from the client to the server. It is therefore the sum of the KPIs mentioned above.

Using the sample Application Monitoring MashApp (available for download in Empower), the total response time is represented in a bar chart in which each single time KPI is reflected in a section of the bar. In the following example, the time KPIs are labeled with the same numbers as in the above graphic. At a glance, you can find out how much response time was spent for each part of the distributed application.



3 Components that Support Application Monitoring

You can monitor distributed application scenarios that make use of the following components:

	z/OS	UNIX	Windows	z/VSE
EntireX Broker [(1)]	x	x	x	x
Listener for XML/SOAP	x	x	x	
EntireX Adapter [(2)]		x	x	
Java RPC Client	x	x	x	
.NET RPC Client			x	
Natural RPC Client	x [(3)]	x	x	x
COBOL RPC Client	x			x
C RPC Client		x	x	
RPC Server for XML/SOAP	x	x	x	
RPC Server for Java	x	x	x	
Natural RPC Server	x [(3)]	x	x	x
RPC Server for CICS	x			x
RPC Server for Batch	x			x
RPC Server for .NET			x	
RPC Server for C		x	x	

Notes:

1. Application monitoring is only supported for the transport methods TCP/IP and SSL.
2. The following connection types of the EntireX Adapter support application monitoring:
   - EntireX RPC Connection
   - EntireX Direct RPC Connection

- EntireX RPC Listener Connection
- EntireX Direct RPC Listener Connection
- IMS Connect Connection
- CICS ECI Connection

3. For z/OS, make sure the EXX load library is part of your steplib chain. We recommend using stub `NATETB23` for all of your Natural RPC environments and a Natural configuration allowing a dynamic load of the stub. This can be achieved by using the following Natural parameters:

   ```
   RCA=(BROKER) RCALIAS=(BROKER,NATETB23)
   ```

   If your broker stub is statically included, you will need to relink your Natural nucleus.

4 Setting Up Application Monitoring

General Information

The collection of response times and other measuring data from your distributed applications in real time requires that you set up various components for application monitoring. EntireX Broker and/or the EntireX Adapter serve as the central components which control the data flow. Their configuration defines the following: whether application monitoring is generally enabled or disabled, the services that are used for monitoring (only for EntireX Broker), and the Application Monitoring Data Collector to which the measuring data is sent.

In addition, your applications need to be prepared for the collection and distribution of measuring data. This is automatically ensured when you use the appropriate EntireX components which support application monitoring. Further configuration on the side of the application is not required.

Setting Up EntireX Broker

EntireX Broker controls the measuring data flow. Using specific attributes in the broker attribute file, the broker can be configured to enable application monitoring for selected services. There are broker-specific and service-specific attributes for application monitoring, and there are also application monitoring-specific attributes. For detailed information, see *Broker Attributes*.

Caution: Changes in the broker attribute file require a restart of the broker. In addition, all involved client applications and RPC servers have to be restarted (after the restart of the broker) because they are caching information about the broker's application monitoring settings.

Configuration example:

```
DEFAULTS = BROKER
  APPLICATION-MONITORING = YES

DEFAULTS = APPLICATION-MONITORING
  COLLECTOR-BROKER-ID = server12:57900

DEFAULTS = SERVICE
  APPLICATION-MONITORING-NAME = Payroll_Application
  CLASS = RPC, SERVER = HR, SERVICE = CALLNAT, APPLICATION-MONITORING = YES, ↩
APPLICATION-MONITORING-NAME = HR_Application
  CLASS = RPC, SERVER = *, SERVICE = CALLNAT, APPLICATION-MONITORING = YES
```

With this example configuration, application monitoring is enabled for all RPC/*/CALLNAT services. The service RPC/HR/CALLNAT uses the application monitoring name "HR_Application", all other services use the name "Payroll_Application". The Application Monitoring Data Collector runs on a host with the name "server12" and uses the port 57900.

Setting Up the EntireX Adapter

The EntireX Adapter automatically supports application monitoring for the following connection types:

- EntireX RPC Connection
- EntireX RPC Listener Connection

If you want to use application monitoring with the following connection types, you have to change the configuration of the EntireX Adapter. See *EntireX Adapter Connections* in the EntireX Adapter documentation.

- EntireX Direct RPC Connection
- EntireX Direct RPC Listener Connection
- IMS Connect Connection
- CICS ECI Connection

Configuration for Application Monitoring

The configuration file *entirex.appmondc.properties* controls the startup of the Application Monitoring Data Collector. It is located in the *config* directory of your EntireX installation.

As a rule, it is not necessary to change the settings in this file after the installation. However, if required, you can change the following parameters:

Parameter	Description
`entirex.appmondc.port`	The TCP/IP port on which the Application Monitoring Data Collector accepts the monitoring data. This value is set during the installation of the Application Monitoring Data Collector.
`entirex.appmondc.directory`	The name of the directory which will contain the CSV data files. The default value is *<EntireX-install-dir>/appmondc/*. A data file has the name *appmon*`<YYYYMMDD>`*.*`<HHMMSS>`*.csv*. In addition, an overview file with the name *appmon.overview.v1.csv* is created.

`entirex.appmondc.loglevel`	The log level for the log files. Possible values are:
	`OFF`
	`FATAL`
	`ERROR`

Parameter	Description
	WARNING INFO DEBUG TRACE The default value is ERROR. Log files are always stored in the directory *<EntireX-install-dir>/appmondc/*. This is independent of the setting of the `entirex.appmondc.directory` parameter.
`entirex.appmondc.maxlines`	The maximum number of rows per CSV data file. If the limit is reached, a new file is created. The default value is 100000.
`entirex.appmondc.filesperday`	Automatically create a new CVS data file every day. The default value is `no`.
`entirex.appmondc.usezeroasnullvalue`	Use "0" instead of an empty entry as the null value for all numeric KPI values in the CSV file. The default value is `no`.
`entirex.appmondc.callback.class`	Java class name for the callback user exit.
`entirex.appmondc.callback.classpath`	Classpath name in URL notation for the callback user exit.

Starting and Stopping the Application Monitoring Data Collector

UNIX

The scripts mentioned below are located in the *bin* directory of your EntireX installation. By default, this is */opt/softwareag/entirex/bin*.

To start the Application Monitoring Data Collector

- Run the script *appmondc.bsh* from a shell.

To stop the Application Monitoring Data Collector

- Run the script *stopappmondc.bsh* from a shell.

Windows

The Application Monitoring Data Collector is installed as an application and as a Windows service. During the installation, you can specify that the Windows service is to be started automatically. The name of this service is "Software AG EntireX Application Monitoring Data Collector 10.1".

Note: You can access the list of services by opening the Start menu and then entering "services.msc" in the search box. There, you can start and stop the service manually.

To start the Application Monitoring Data Collector as an application

- Choose the following from the Windows Start menu:

  Programs > Software AG > Start Servers > Start EntireX Application Monitoring Data Collector 10.1

  Or:

  Run the script *appmondc.bat* which is located in the *bin* directory of your EntireX installation.

To stop the Application Monitoring Data Collector (service and application)

- Choose the following from the Windows Start menu:

  Programs > Software AG > Stop Servers > Stop EntireX Application Monitoring Data Collector 10.1

  Or:

  Run the script *stopappmondc.bat* which is located in the *bin* directory of your EntireX installation.

Callback User Exit

The Application Monitoring Data Collector provides a callback functionality for the processing of incoming events. A user exit can be specified to implement the callback. Whenever the Data Collector receives a monitoring event, the KPI values are writen to the CSV file and the callback is invoked. The KPI values of the event are passed to the callback. Possible use cases are:

- trigger an action based on specific KPI values (e.g. the KPIs indicate a failed request)
- write the KPI values to another data store or in a format that is different from a CSV file

To enable the user exit callback, use the property `entirex.appmondc.callback.class` to specify the class name of the user exit implementation. The class is loaded using the standard classpath. You can specify a separate classpath with the property `entirex.appmondc.callback.classpath`.

Note that for the classpath, a file or HTTP URL must be specified (on Windows replace the "\" character with "/"). Your user exit class must implement the Java interface `com.softwareag.entirex.appmondc.DataCollectorCallback`. This Java interface has the following methods:

```
/**
 * Initialize the callback handler.
 * @param directory The name of the directory that will contain the CSV data files ↩
(set by property "entirex.appmondc.directory")
 *
 * @throws Exception
 */
public void start(String directory) throws Exception;

/**
 * Stop the callback handler.
 */
public void stop();

/**
 * Process an event. This corresponds to an entry in the CSV file.
 * The map contains all KPIs which have a value.
 * The key names of the map are identical to the KPI names.
 *
 * @param kpis The KPI map.
 * @throws Exception
 */
public void processEvent(Map<String, String> kpis) throws Exception;
```

5 KPI Definitions for Application Monitoring

General Information

The tables below describe the KPIs (key performance indicators) monitored by the Application Monitoring Data Collector. Each KPI is represented as a row in the CSV file produced by the data collector. The KPI name is identical to the row name in the first column of the CSV file. There is only one common layout of the CSV file. Therefore, depending on the scenario, KPIs may have no values. In this case, the column has an empty entry in the corresponding row. This can be changed to the value "0" by setting the parameter `entirex.appmondc.usezeroasnullvalue` in the configuration file *entirex.appmondc.properties* (see ***Configuration for Application Monitoring***).

There are three different scenarios: RPC, CICS ECI and IMS Connect. Each scenario has a different set of KPIs for successful requests and for failed requests. For a successful request, the KPI "ErrorCode" is always empty. For a failed request this KPI always has a value.

The RPC scenario is supported when using the EntireX Broker as well as when using the *Direct RPC* component of the EntireX Adapter. The CICS ECI and IMS Connect scenarios are supported by the EntireX Adapter only when using the corresponding connection types.

The sequence of the KPIs in the tables below is the same as the sequence of the KPIs in the CSV file.

KPIs for RPC - Successful Requests

KPI Name	Description
Time	The time the event has been processed by the data collector in the format "YYYY-MM-DD HH:MM:SS.SSS" using the current time zone.
Timestamp	The time the event has been processed by the data collector as a number. The number is the difference, measured in milliseconds, between the current time and midnight, January 1, 1970 UTC.
Scenario	The scenario identifier "RPC".
ApplicationName	"*`application-name`*" as defined by the broker attribute `APPMON-NAME`. If the broker attribute is not specified, the server address is used; for example `RPC/SRV/CALLNAT`.
Address	The broker ID and the server address of the RPC request.
TimeResponse	The complete response time (roundtrip from client to server and back) in microseconds.
TimeClientLayer	The time spent in the client RPC layer in microseconds.
TimeClientTransport	The transport time from the client to the broker and back in microseconds.
TimeBroker	The time spent in the broker (active processing) in microseconds.
TimeBrokerWaitForServer	The time spent in the broker waiting for an available server in microseconds.

KPI Name	Description
TimeServerTransport	The transport time from the broker to the server and back in microseconds.
TimeServerLayer	The time spent in the server RPC layer (runtime and stub) in microseconds.
TimeServerProgram	The time spent in the user program (called by the RPC server) in microseconds. For Natural programs on a mainframe, this time does not include the database times. For other programs, the database times are included.
TimeDBCalls	The time spent for database calls in microseconds. For an Adabas database, this is the time the Adabas server needs to process the database call ("client wait time"). For other databases, the DB calls time includes also the DB transport time. [1]
TimeDBTransport	The transport time from the Natural user program to the Adabas router and back including the client receiving time in microseconds. [1, 2]
Program	The program name.
ClientApplication	The client application name as defined in the broker control block.
ClientHost	The client host name.
ClientUser	The client user ID.
LengthRequest	The length of the RPC request in bytes.
LengthReply	The length of the RPC reply in bytes.
LengthTotal	The total length of the RPC call (request plus reply) in bytes.
DBCalls	The number of database calls (including system file calls, without Natural Security calls). [1]
ErrorCode	Always empty.

Notes:

[1] This KPI is only available if the call is issued by a Natural RPC server on a mainframe.

[2] This KPI is only available for a database call against an Adabas server.

KPIs for RPC - Failed Requests

KPI Name	Description
Time	The time the event has been processed by the data collector in the format "YYYY-MM-DD HH:MM:SS.SSS" using the current time zone.
Timestamp	The time the event has been processed by the data collector as a number. The number is the difference, measured in milliseconds, between the current time and midnight, January 1, 1970 UTC.
Scenario	The scenario identifier "RPC".
ApplicationName	"*application-name*" as defined by the broker attribute APPMON-NAME. If the broker attribute is not specified, the server address is used; for example RPC/SRV/CALLNAT.

KPI Name	Description
Address	The broker ID and the server address of the RPC request.
TimeResponse	The response time of the failed RPC request in microseconds.
Program	The program name.
ClientApplication	The client application name as defined in the broker control block.
ClientHost	The client host name.
ClientUser	The client user ID.
ErrorCode	The 8-digit error code (error class and number).
ErrorMessage	The error message.

KPIs for CICS ECI - Successful Requests

KPI Name	Description
Time	The time the event has been processed by the data collector in the format "YYYY-MM-DD HH:MM:SS.SSS" using the current time zone.
Timestamp	The time the event has been processed by the data collector as a number. The number is the difference, measured in milliseconds, between the current time and midnight, January 1, 1970 UTC.
Scenario	The scenario identifier "CICS ECI".
ApplicationName	"*host-name*:*port-number*" of the CICS ECI installation.
Address	The name of the Integration Server adapter service which calls CICS ECI.
TimeResponse	The complete response time of the CICS ECI request in microseconds.
TimeClientLayer	The time spent in the EntireX Adapter in microseconds.
TimeServerLayer	The sum of the transport time to CICS ECI and the time spent in the CICS user program in microseconds.
Program	The CICS transaction name.
ClientHost	The client host name.
ClientUser	The client user ID.
LengthRequest	The length of the CICS request in bytes.
LengthReply	The length of the CICS reply in bytes.
LengthTotal	The total length of the CICS call (request plus reply) in bytes.
ErrorCode	Always empty.

KPIs for CICS ECI - Failed Requests

KPI Name	Description
Time	The time the event has been processed by the data collector in the format "YYYY-MM-DD HH:MM:SS.SSS" using the current time zone.
Timestamp	The time the event has been processed by the data collector as a number. The number is the difference, measured in milliseconds, between the current time and midnight, January 1, 1970 UTC.
Scenario	The scenario identifier "CICS ECI".
ApplicationName	"*host-name*:*port-number*" of the CICS ECI installation.
Address	The name of the Integration Server adapter service which calls CICS ECI.
TimeResponse	The response time of the failed CICS ECI request in microseconds.
Program	The CICS transaction name.
ClientHost	The client host name.
ClientUser	The client user ID.
ErrorCode	The 8-digit error code (error class and number).
ErrorMessage	The error message.

KPIs for IMS Connect - Successful Requests

KPI Name	Description
Time	The time the event has been processed by the data collector in the format "YYYY-MM-DD HH:MM:SS.SSS" using the current time zone.
Timestamp	The time the event has been processed by the data collector as a number. The number is the difference, measured in milliseconds, between the current time and midnight, January 1, 1970 UTC.
Scenario	The scenario identifier "IMS Connect".
ApplicationName	"*host-name*:*port-number*/*datastore*" of the IMS Connect installation.
Address	The name of the Integration Server adapter service which calls IMS Connect.
TimeResponse	The complete response time of the IMS request in microseconds.
TimeClientLayer	The time spent in the EntireX Adapter in microseconds.
TimeServerLayer	The sum of the transport time to IMS Connect and the time spent in IMS Connect, IMS and the IMS user program in microseconds.
Program	The IMS transaction name.
ClientHost	The client host name.
ClientUser	The client user ID.

KPI Name	Description
LengthRequest	The length of the IMS request in bytes.
LengthReply	The length of the IMS reply in bytes.
LengthTotal	The total length of the IMS call (request plus reply) in bytes.
ErrorCode	Always empty.

KPIs for IMS Connect - Failed Requests

KPI Name	Description
Time	The time the event has been processed by the data collector in the format "YYYY-MM-DD HH:MM:SS.SSS" using the current time zone.
Timestamp	The time the event has been processed by the data collector as a number. The number is the difference, measured in milliseconds, between the current time and midnight, January 1, 1970 UTC.
Scenario	The scenario identifier "IMS Connect".
ApplicationName	"*host-name*:*port-number*/*datastore*" of the IMS Connect installation.
Address	The name of the Integration Server adapter service which calls IMS Connect.
TimeResponse	The response time of the failed IMS request in microseconds.
Program	The IMS transaction name.
ClientHost	The client host name.
ClientUser	The client user ID.
ErrorCode	The 8-digit error code (error class and number).
ErrorMessage	The error message.