

webMethods EntireX

Application Monitoring

Version 9.7

October 2014

This document applies to webMethods EntireX Version 9.7.

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

Copyright © 1997-2014 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA, Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at <http://softwareag.com/licenses>.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at <http://softwareag.com/licenses/> and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third-Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at <http://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

Use, reproduction, transfer, publication or disclosure is prohibited except as specifically provided for in your License Agreement with Software AG.

Document ID: EXX-APPMON-97-20160805

Table of Contents

Preface	v
1 Introduction	1
What is Application Monitoring?	2
Sample Scenarios	2
Response Time KPIs	4
2 Components Which Support Application Monitoring	7
3 Setting Up Application Monitoring	9
General Information	10
Setting Up EntireX Broker	10
Setting Up the EntireX Adapter	11
Configuration for Application Monitoring	11
Starting and Stopping the Application Monitoring Data Collector	12
4 KPI Definitions for Application Monitoring	15
General Information	16
KPIs for RPC - Successful Requests	16
KPIs for RPC - Failed Requests	17
KPIs for CICS ECI - Successful Requests	18
KPIs for CICS ECI - Failed Requests	19
KPIs for IMS Connect - Successful Requests	19
KPIs for IMS Connect - Failed Requests	20
5 Application Monitoring MashApp	21
General Information	22
Preparing to Use the MashApp	22
Special Configuration for Application Monitoring	25
Using the MashApp	26

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 Which 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.
Application Monitoring MashApp	Information on the sample MashApp that is available for application monitoring.

1 Introduction

- What is Application Monitoring? 2
- Sample Scenarios 2
- Response Time KPIs 4

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 also provided.

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

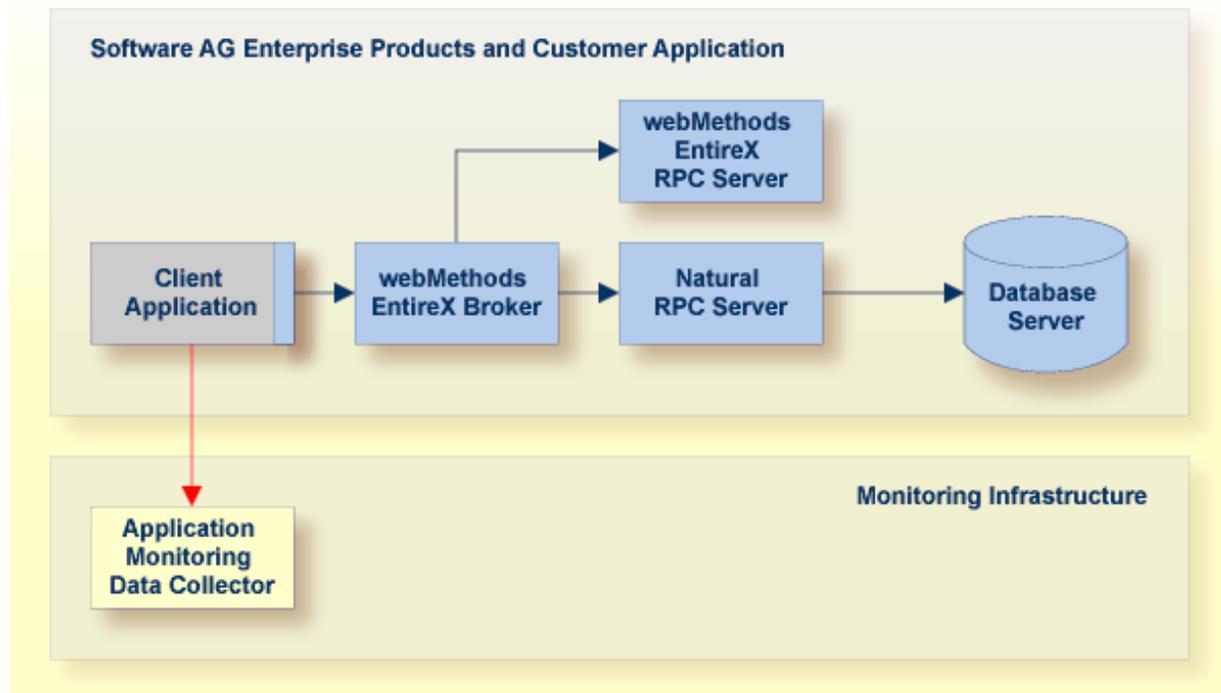
The following graphics illustrate examples of different scenarios in which application monitoring can be used. The boxes in the graphics contain the components that are running in your production environment. The components in the blue boxes can be monitored. The components in the gray boxes cannot be 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 Which Support Application Monitoring](#).

RPC Scenario with EntireX Broker

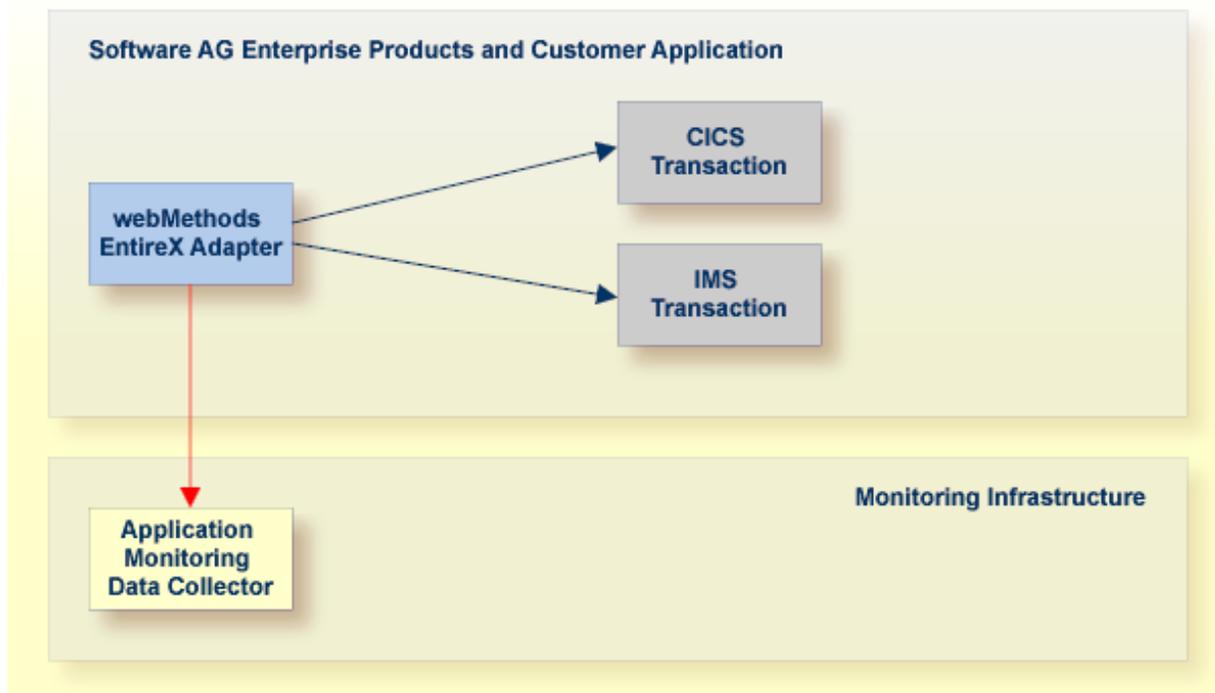
The small blue box linked to the gray "Client Application" box represents the client RPC layer which can be monitored, whereas the client application itself cannot be monitored. Note that the time spent for database calls can only be monitored if the call is issued by a Natural RPC server.



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

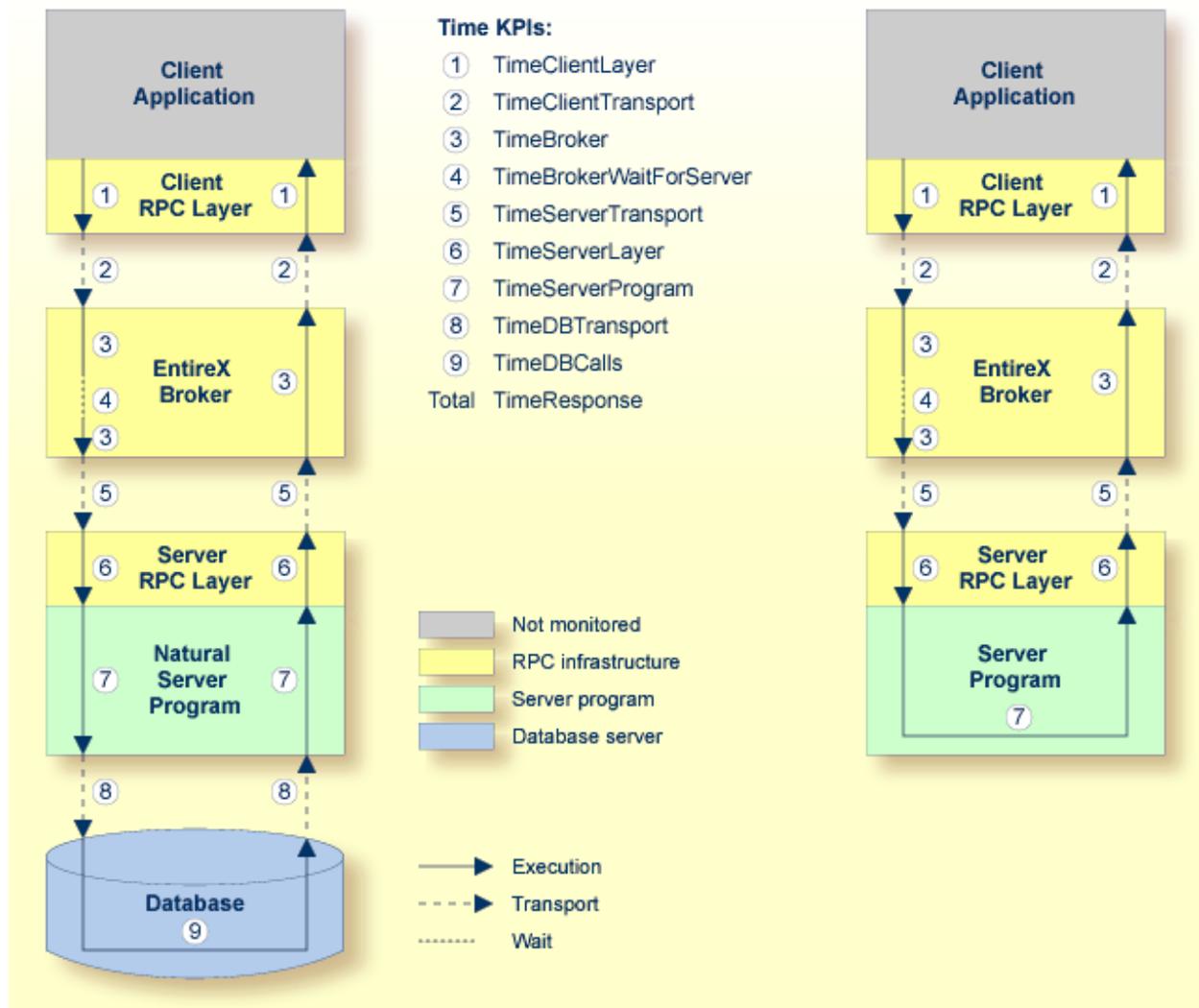
CICS ECI and IMS Connect Scenario with EntireX Adapter

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.



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, 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.



2 Components Which 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
XML/Soap Listener	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	
XML/Soap RPC Server	X	X	X	
Java RPC Server	X	X	X	
Natural RPC Server	X ³	X	X	X
CICS RPC Server	X			X
Batch RPC Server	X			X
.NET RPC Server			X	
C RPC Server		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.

3

Setting Up Application Monitoring

- General Information 10
- Setting Up EntireX Broker 10
- Setting Up the EntireX Adapter 11
- Configuration for Application Monitoring 11
- Starting and Stopping the Application Monitoring Data Collector 12

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 the EntireX Adapter documentation for further information):

- 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
<code>entirex.appmondc.port</code>	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.
<code>entirex.appmondc.directory</code>	The name of the directory which will contain the CSV data files. The default value is <code><EntireX-install-dir>/appmondc/</code> . A data file has the name <code>appmon<YYYYMMDD>.<HHMMSS>.csv</code> . In addition, an overview file with the name <code>appmon.overview.v1.csv</code> is created.
<code>entirex.appmondc.loglevel</code>	The log level for the log files. Possible values are: OFF FATAL ERROR

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

 **Note:** If you plan to use the Application Monitoring MashApp, some of the above parameters require special settings. For further information, see [Special Configuration for Application Monitoring](#).

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 start script `appmondc.bsh` from a shell.

➤ To stop the Application Monitoring Data Collector

- Run the start 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 9.7".



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 9.7

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 9.7

Or:

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

4 KPI Definitions for Application Monitoring

- General Information 16
- KPIs for RPC - Successful Requests 16
- KPIs for RPC - Failed Requests 17
- KPIs for CICS ECI - Successful Requests 18
- KPIs for CICS ECI - Failed Requests 19
- KPIs for IMS Connect - Successful Requests 19
- KPIs for IMS Connect - Failed Requests 20

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 "Error-Code" 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	" <i>application-name</i> " 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	" <i>application-name</i> " 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	<i>"host-name:port-number"</i> 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	<i>"host-name:port-number"</i> 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	<i>"host-name:port-number/datastore"</i> 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	" <i>host-name:port-number/datastore</i> " 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.

5 Application Monitoring MashApp

- General Information 22
- Preparing to Use the MashApp 22
- Special Configuration for Application Monitoring 25
- Using the MashApp 26

General Information

The Application Monitoring Data Collector collects the response time data of each involved software component of selected synchronous EntireX RPC, CICS ECI or IMS Connect services and stores it in CSV (comma-separated values) files.

MashZone is a browser-based application from Software AG which enables you to visualize any data from various, independently distributed data sources in a so-called MashApp. Software AG provides a sample Application Monitoring MashApp which shows and evaluates the data which is collected by the Application Monitoring Data Collector. It is a showcase how a KPI-specific visualization can be achieved with MashZone.

The Application Monitoring MashApp supports MashZone 2.3 and above.

Preparing to Use the MashApp

The following topics are covered below:

- [Downloading the MashApp](#)
- [Unpacking the Zip File](#)
- [Editing the Resource Files](#)
- [Activating the MashApp](#)

Downloading the MashApp

The Application Monitoring MashApp and related data are provided in a zip file in Empower, as a download component for EntireX. To download the zip file, proceed as follows:

1. Log in to Empower (<https://empower.softwareag.com/>).
2. Go to **Products & Documentation > Download Components > EntireX - Application Monitoring MashApp**.
3. Download the "Application Monitoring MashApp" component (*EXX_AppMon_MashApp.zip*).

In addition to the zip file, Empower also provides a readme file (*Readme_EXX_AppMon_MashApp.txt*) which contains the latest update information.

Unpacking the Zip File

You have to unpack the *EXX_AppMon_MashApp.zip* file to the following MashZone user data directory, depending on the MashZone version that you are using:

- For MashZone 9.0 and above:

```
<installation-directory>\server\bin\work\work_mashzone_<t>\mashzone_data
```

<t> indicates the type of the MashZone server. Types are *s*, *m*, and *l*. For example, *work_mashzone_m* for a medium type.

- For MashZone versions lower than 9.0:

```
<installation-directory>
```

where <installation-directory> is the MashZone installation directory.

After unpacking the zip file, the following directories are available in the MashZone user data directory:

Directory	Content
<i>importexport\ApplicationMonitoring_<date></i>	MashApp for application monitoring.
<i>resources\ApplicationMonitoring\Definition</i>	Resources used by the MashApp. Initially, this directory contains the resources which do not have to be edited.
<i>resources\ApplicationMonitoring\Definition_src</i>	Source directory for resources which have to be edited (further information is provided below) and copied into the <i>resources\ApplicationMonitoring\Definition</i> directory.
<i>resources\ApplicationMonitoring\Data</i>	Default application monitoring directory. This is the directory where the MashApp expects by default the application monitoring overview and data files.
<i>assets\colorschemes</i>	Color schemes. The color schemes for application monitoring are named <i>AppMon_*.xml</i> .

Editing the Resource Files

To adapt the Application Monitoring MashApp to your requirements, you have to edit the resource files in the *resources\ApplicationMonitoring\Definition_src* directory as described below. The resource files are CSV files with commas (,) as separators. To edit them, any tool which is able to handle text files can be used.

The following resource files are provided:

■ *Directory.csv*

The Application Monitoring MashApp expects that the application monitoring CSV files are available in any subdirectory of the `<MashZone-user-data>\resources\ApplicationMonitoring` directory. If the Application Monitoring Data Collector writes its data directly into such a subdirectory, you can continuously monitor the incoming data. If your application monitoring data has been written into any other directory, you can manually copy the whole directory (including the *appmon.overview.v1.csv* file) into a subdirectory of the `<MashZone-user-data>\resources\ApplicationMonitoring` directory and analyze the data with the MashApp. The MashApp refreshes every minute so that it always displays the most recent data.

The *Directory.csv* file contains the relative names of the subdirectories containing the application monitoring data. If you use multiple directories, you have to specify one line for each directory.

The first line in the file is in general used as the default directory when you start the MashApp. Further directory entries can be selected on the **Data Files** tab of the MashApp. The delivered *Directory.csv* file contains one sample line for the *Data* directory. If you do not modify the *Directory.csv* file, all application monitoring data files are expected in the `resources\ApplicationMonitoring\Data` directory.

Column	Description
Enable	An entry is only shown on the Data Files tab of the MashApp if "Enable" is set to "y".
Description	Specify a descriptive name for the directory. The descriptions are used in the data file selection of the MashApp. If the description is empty, the "Path" value is used as the description.
Path	Specify the path to the application monitoring directory. The path is always left-expanded with <code>./resources/ApplicationMonitoring/</code> . The default path is <i>Data</i> which is expanded to <code>./resources/ApplicationMonitoring/Data</code> .

■ *Properties.csv*

This file contains settings for the Application Monitoring MashApp. You can specify a value for each given property. If you do not modify the *Properties.csv* file, the default property values are used.

Property	Description
Response Time yellow	The response time reflects the complete time of the roundtrip from the client to the server and back. The lower the response time, the better the overall application performance. Good, average or poor performance is indicated in the MashApp by a green, yellow or red colored background, respectively. Specify the value of the yellow threshold (in units of milliseconds) for which the response time leaves the green zone. Default: 1000.
Response Time red	Specify the value of the red threshold (in units of milliseconds) for which the response time leaves the yellow zone. Default: 2000.

Activating the MashApp

To activate the Application Monitoring MashApp, make sure you have a Professional, Enterprise or Event license file, that you have administrator rights, and then proceed as follows:

1. Copy all resource files from *resources\ApplicationMonitoring\Definition_src* to *resources\ApplicationMonitoring\Definition*.
2. Invoke MashZone.
3. Go to the **Administration** page (see the corresponding link at the top of the page) and then to the **Import/Export/Delete** page.
4. Import the MashZone archive file (*.mzp) from the *importexport\ApplicationMonitoring_<date>* directory using the **Import** function.

The MashApp in the *importexport\ApplicationMonitoring_<date>* directory is named as follows:

M_Application Monitoring <version>_<revision>_<date>-<time>.mzp

Special Configuration for Application Monitoring

The following configuration parameters of the Application Monitoring Data Collector (see [Configuration for Application Monitoring](#)) require special settings when data is to be evaluated by the Application Monitoring MashApp.

Parameter	Special Configuration
entirex.appmondc.directory	<p>If the MashApp is to continuously monitor the application monitoring data, set this parameter to</p> <p><i><MashZone-user-data>/resources/Application-Monitoring/Data</i></p> <p>Alternatively, you can also use a different subdirectory of the <i>resources/ApplicationMonitoring</i> directory. In this case, you have to adjust the <i>Directory.csv</i> resource file accordingly.</p>
entirex.appmondc.maxlines	<p>MashZone Version 9.6 and above is able to process a maximum of 100000 lines in a table. For previous versions, the maximum value is 10000 lines.</p> <p>By default, the MashApp concatenates seven CSV files. It is therefore recommended to restrict this parameter to 14000 for MashZone Version 9.6 and above, or to 1400 for previous versions. If you plan to concatenate less CSV files in general, you can use a higher value.</p>
entirex.appmondc.usezeroasnullvalue	Set this parameter to <i>yes</i> to avoid errors in the MashApp.

Using the MashApp

After you have specified all required information as described above, you can proceed as follows:

1. Invoke MashZone.
2. Open the MashApp "Application Monitoring".
3. Select the **Overview** tab.



The MashApp consists of several tabs offering various functions for monitoring and analyzing the application monitoring data. It also has one tab for data file selection, and one tab for the overview. The overview page shows a brief description of the tabs and charts available in the MashApp (see also the following table) and provides links to the corresponding functions.

Tab Name	Function Name	Description
Response Time	Response Time Monitor	Provides history charts which show the general performance and system load consumed by your distributed application.
Transactions	Transaction Monitor	Provides history charts which show the general service usage.
Successful	Successful Transactions	Provides a response time analysis for successful transactions.
Error	Error Transactions	Provides a response time analysis for error transactions.
Average	Average Response Time	Provides a performance analysis for the components of your distributed application.
Total	Total Response Time	Provides an analysis of the system load consumed by the components of your distributed application.
Distribution	Distribution	Shows the distribution of the selected KPI (transactions, response times, database calls) over the instances of selected components (scenario, application, program, client host, client application, or client user).
High Values	High Values	Shows for each component the most used instances, maximum response times, and more.
Aggregation	Aggregation	Shows summarized information for all instances of the selected aggregation criterion.
Data Files	Data File Selection	Allows you to select the CSV data files that are to be evaluated by the MashApp.
Overview	Application Monitoring Overview	Gives an overview of the functions and charts of the MashApp, including links to the corresponding tabs.

The following topics are covered below:

- Common Master View
- Response Time Monitor
- Transaction Monitor
- Successful Transactions
- Error Transactions
- Average Response Time
- Total Response Time
- Distribution
- High Values
- Aggregation
- Data File Selection
- Application Monitoring Overview

Common Master View

On most tabs, a common master view is shown at the top.

Transaction Monitor 2014-06-28 17:08 - 18:08		Time Range in the Selected Files	Scenario	Client Application
		Last hour	All scenarios	All client applications
Number of Transactions		Only Valid for Custom Range	Application in Data Collector	Client User
		06/26/2014 OK 00 00	All applications	All client users
Success	120	Only Valid for Custom Range	Server Program	
Error	0	06/28/2014 OK 23 59	All programs	
Total	120	Time Interval	Client Host	
		1 minute	All client hosts	

In general, the master view contains the following elements (from left to right, and from top to bottom):

- The MashApp function (for example, "Transaction Monitor") and the currently selected time range.
- Summarized data for the current function (for example, the number of transactions) according to the values that are currently selected in the master view.
- A selection box for selecting the time range. The following predefined time ranges are available:

Last hour
 Last 2 hours
 Last 4 hours
 Last 8 hours
 Last day
 Last 2 days
 Last week
 Last 2 weeks
 Last month
 Last quarter
 Last year
 All data in the selected files

Custom range

The **Last ...** time ranges always use the last entry in the selected data files as the basis. For example, if the last entry in the selected data files was written at 09:00, the **Last hour** refers to the time from 8:00 to 9:00.

The MashApp offers only those **Last ...** time ranges which are appropriate for the currently selected data files. For example, if the selected data files comprise 3 hours, the MashApp will not offer **Last 4 hours** or higher values.

The time range **All data in the selected files** refers to all data in the data files that are currently selected on the **Data Files** tab.

When you select **Custom range**, the titles for the selection boxes in the next two lines will change from **Only Valid for Custom Range** to **Custom Range From** and **Custom Range To**. You can then specify the desired start and end times (date, hour and minute) for the time range in these two lines. The custom range is not restricted to the data of the currently selected data files. It always reads from the start date until it reaches the end date, or until seven data files have been read (the MashApp is only able to evaluate up to seven data files).

- A selection box for the time interval. The following predefined time intervals are available:

- 1 minute
- 5 minutes
- 10 minutes
- 15 minutes
- 30 minutes
- 1 hour
- 2 hours
- 4 hours
- 8 hours
- 1 day
- 1 week

The time interval is used when the data has to be aggregated on the time axis of a history chart. For example, if the time interval is **10 minutes**, all data of a 10 minute time interval is aggregated into one value which is displayed in the history chart.

The MashApp offers only those time intervals which are appropriate for the currently selected time range and which will not end up with too many entries. For example, if the selected time range comprises 1 week, the MashApp will offer the aggregated time intervals from 2 hours to 1 day.

- Selection boxes for the following components:

- Scenario
- Application in Data Collector
- Server Program

Client Host
Client Application
Client User

The charts on the different tabs of the MashApp evaluate the data according to the selected components. By default, the MashApp evaluates all data for all components.

The MashApp only offers for selection the components which are available in the currently selected data files and in the currently selected time range. The component selection boxes are filled from top to bottom. The content of each selection box depends on the selection in the previous selection box.

Possible scenarios are:

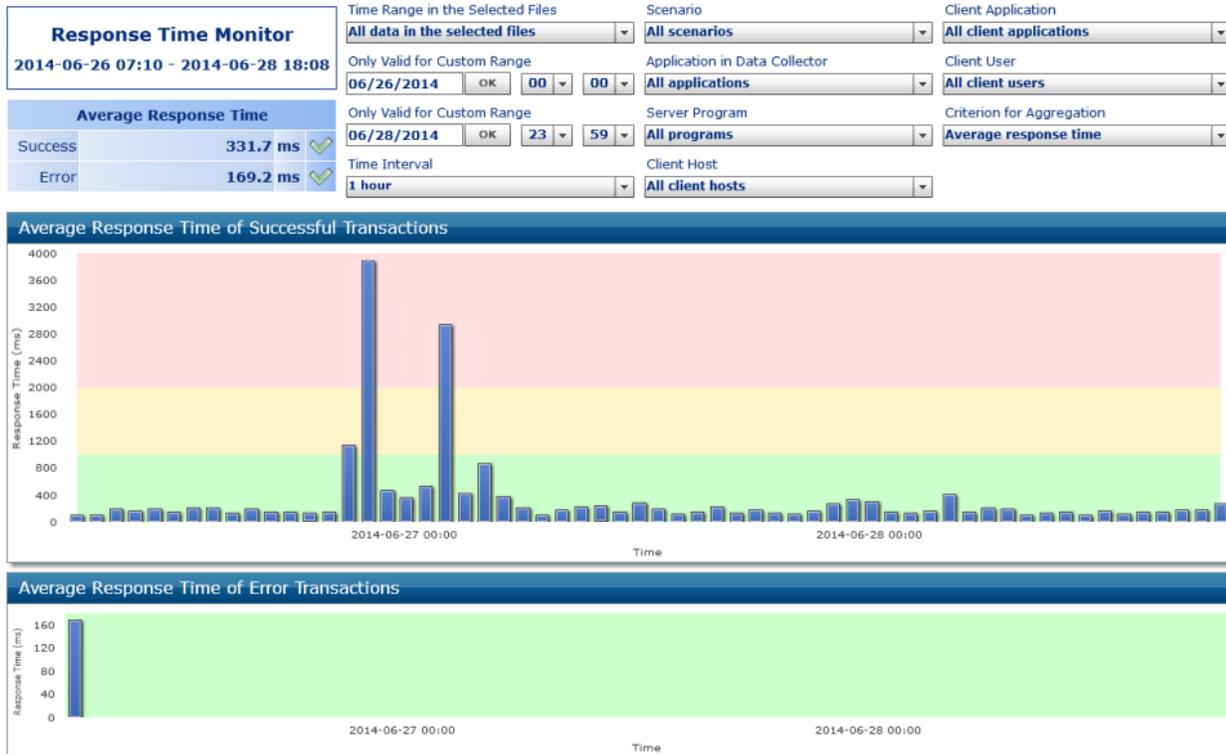
RPC
CICS ECI
IMS Connect

The application names are defined in the EntireX broker attribute file. See [Setting Up EntireX Broker](#).

- Other selection boxes in the common master view, which have not been mentioned above, are function-specific and are described in the corresponding sections later in this documentation.

Response Time Monitor

The **Response Time** tab shows the history of the response time according to the selected time range and components. The data is aggregated according to the selected aggregated time interval.



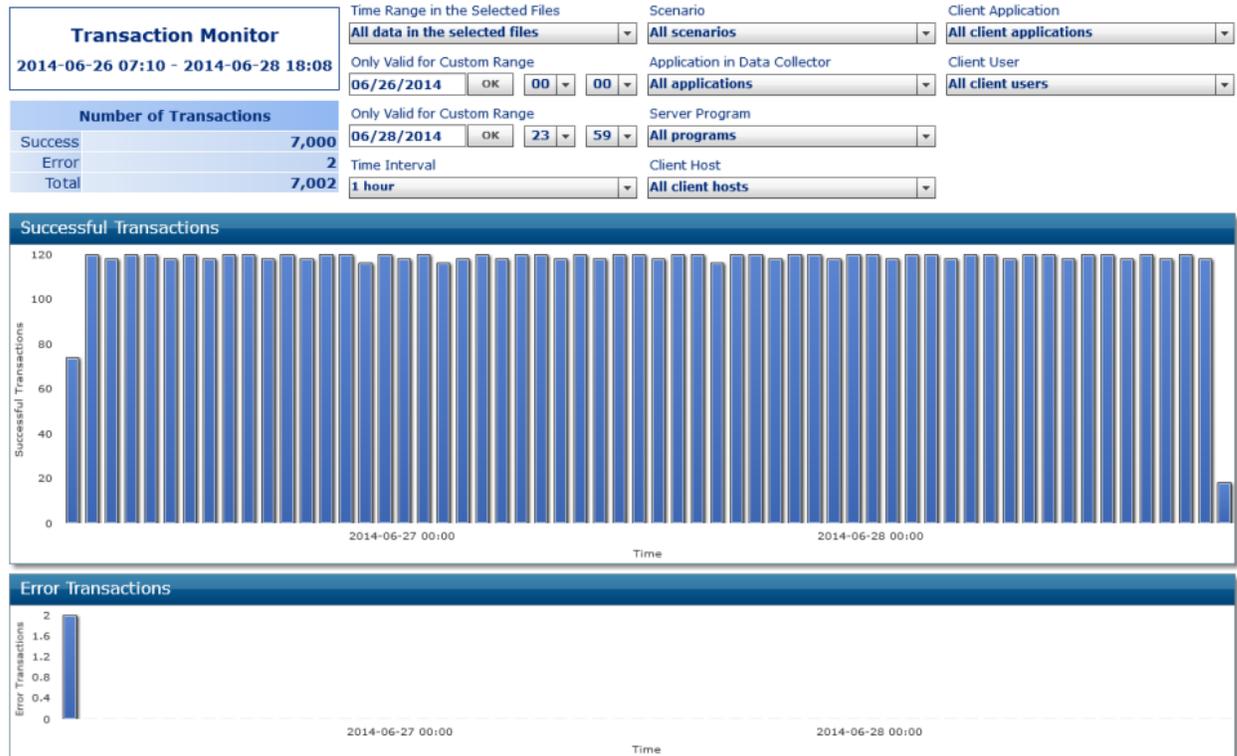
The upper chart shows the history of successful transactions; the lower chart shows the history of error transactions. You can select the following from the **Criterion for Aggregation** selection box:

- **Average response time**
The average response time is an indicator for the performance of your distributed application. The lower the average response time, the better the performance. Good, average or poor performance is indicated by the colored background according to the settings in the *Properties.csv* file.
- **Maximum response time (default)**
The maximum response time shows the response time peaks of your distributed application for each aggregated time interval. The chart helps to identify single transactions which perform poorly, regardless of how the application has performed in average. Good, average or poor performance is indicated by the colored background according to the settings in the *Properties.csv* file.
- **Total response time**
The total response time is an indicator for the system load consumed by your distributed application. Even an application with a good performance may have a high system load if it is called very often.

The summarized data at the top of the tab shows the average, maximum or total response time for the selected successful transactions and error transactions. For the average and maximum response times, icons indicate good, average or poor performance according to the settings in the *Properties.csv* file.

Transaction Monitor

The **Transactions** tab shows the history of the number of transactions according to the selected time range and components. It is an indicator for the general service usage. The data is aggregated according to the selected aggregated time interval.



The upper chart shows the history of successful transactions; the lower chart shows the history of error transactions.

The summarized data at the top of the tab shows the total number of successful transactions and error transactions, and the total of both for the selected time range and components.

Successful Transactions

The **Successful** tab shows the successful transactions in the selected time range for the selected components.

Successful Transactions

2014-02-28 14:47 - 2014-04-04 14:51

Transactions	376
Avg Rsp Time	971.9 ms
Max Rsp Time	6,749.6 ms
Total Rsp Time	365,456.0 ms

Time Range in the Selected Files

All data in the selected files

Only Valid for Custom Range

06/26/2014 OK 00 00

Only Valid for Custom Range

06/28/2014 OK 23 59

Scenario

All scenarios

Application in Data Collector

All applications

Server Program

All programs

Client Application

All client applications

Client User

All client users

Client Host

All client hosts

Recording Time (local)	Scenario	Application (Data Collector)	Program (Server)	Host (Client)	Application (Client)	User (Client)	Response Time [ms]	Client Layer Time [ms]	Client Transport Time [ms]
2014-03-04 14:41:03	RPC	RPC/SRV1/CALLNAT	SQUARE	MCDR02	IDL Tester	drb	705.200	180.000	
2014-03-04 14:41:03	RPC	RPC/SRV1/CALLNAT	SQUARE	MCDR02	IDL Tester	drb	778.000	195.200	
2014-03-04 14:41:03	RPC	RPC/SRV1/CALLNAT	SQUARE	MCDR02	IDL Tester	drb	670.000	169.600	
2014-03-04 14:41:02	RPC	RPC/SRV1/CALLNAT	SQUARE	MCDR02	IDL Tester	drb	709.200	175.200	
2014-03-04 14:41:02	RPC	RPC/SRV1/CALLNAT	SQUARE	MCDR02	IDL Tester	drb	824.000	244.800	
2014-03-04 14:41:01	RPC	RPC/SRV1/CALLNAT	SQUARE	MCDR02	IDL Tester	drb	1,346.400	743.600	
2014-03-04 14:41:01	RPC	RPC/SRV1/CALLNAT	SQUARE	MCDR02	IDL Tester	drb	678.000	190.000	
2014-03-04 14:41:00	RPC	RPC/SRV1/CALLNAT	SQUARE	MCDR02	IDL Tester	drb	4,724.400	85.200	
2014-02-28 16:40:07	RPC	RPC/SRV1/CALLNAT	TEST	MCLHU	Rand001	lukas	1,195.200	248.800	
2014-02-28 16:39:07	RPC	RPC/SRV1/CALLNAT	TEST	MCLHU	Rand001	lukas	1,191.600	248.400	
2014-02-28 16:38:07	RPC	RPC/SRV1/CALLNAT	TEST	MCLHU	Rand001	lukas	1,188.000	248.000	
2014-02-28 16:37:07	RPC	RPC/SRV1/CALLNAT	TEST	MCLHU	Rand001	lukas	1,184.400	247.600	
2014-02-28 16:36:07	RPC	RPC/SRV1/CALLNAT	TEST	MCLHU	Rand001	lukas	1,180.800	247.200	
2014-02-28 16:35:07	RPC	RPC/SRV1/CALLNAT	TEST	MCLHU	Rand001	lukas	1,177.200	246.800	
2014-02-28 16:34:07	RPC	RPC/SRV1/CALLNAT	TEST	MCLHU	Rand001	lukas	1,173.600	246.400	

Transaction ID: 1000324
Recording Time: 2014-02-28 16:40:07
Response Time: 1,195.200ms
Address: daeexx23/RPC/SRV1/CALLNAT

Scenario: RPC
Application: RPC/SRV1/CALLNAT
Program: TEST

Client Host: MCLHU
Client Application: Rand001
Client User: lukas

RPC Call Length Total: 517
RPC Call Len. Request: 317
RPC Call Length Reply: 312
DB Calls: 140

The table shows the application monitoring KPIs for each transaction. These are the KPIs for the successful requests as described in [KPI Definitions for Application Monitoring](#).

In addition to the application monitoring KPIs, the table lists the **Transaction ID** which uniquely identifies each transaction. The transaction ID consists of the sequence number of the currently used data file in the *appmon.overview.v1.csv* file and (for the last six digits) the number of the record in the data file. By default, the transactions in the table are sorted by the transaction ID. You can change the sorting by clicking on the header of any column.

The **Response Time** column is colored according to the settings in the *Properties.csv* file, indicating good, average or poor performance.

When you select (click) a transaction, all available column values are shown at the bottom of the tab and the distribution of the measured times is visualized in the right bar chart below the transaction table. The left bar chart shows the response time of the currently selected transaction and compares it with the minimum, average and maximum response times of all transactions in the table. By default (if no transaction is selected), both charts show the average response time and the background is colored according to the settings in the *Properties.csv* file, indicating good, average or poor performance.

The summarized data at the top of the tab shows the number of successful transactions and the average, maximum and total response times for all successful transactions in the selected time range for the selected components. For the average and maximum response times, icons indicate good, average or poor performance according to the settings in the *Properties.csv* file.

32

Application Monitoring

Error Transactions

The **Error** tab shows the error transactions in the selected time range for the selected components.

Error Transactions

2014-02-28 14:47 - 2014-04-04 14:51

Transactions	8
Avg Rsp Time	3,493.5 ms
Max Rsp Time	9,096.8 ms
Total Rsp Time	27,948.4 ms

Time Range in the Selected Files
All data in the selected files

Only Valid for Custom Range
06/26/2014 OK 00 00

Only Valid for Custom Range
06/28/2014 OK 23 59

Scenario
All scenarios

Application in Data Collector
All applications

Server Program
All programs

Client Host
All client hosts

Client Application
All client applications

Client User
All client users

Cumulated Errors					
Error Code	Occurrence	Avg Rsp Time [ms]	Max Rsp Time [ms]	Total Rsp Time [ms]	Error Message
20120353	2	9,096.800	9,096.800	18,193.600	java.net.UnknownHostException: a
Rc 0007	2	454.000	454.000	908.000	RPC/SRV1/CALLNAT not registered
00070007	2	1,541.600	2,217.600	3,083.200	RPC/SRV1/CALLNATx not registered
Rsp 0007	2	2,881.800	4,865.600	5,763.600	RPC/SRV1/CALLNATx not registered

Error Transactions									
Recording Time (local)	Scenario	Application (Data Collector)	Program (Server)	Host (Client)	Application (Client)	User (Client)	Response Time [ms]	Error Code	Error Message
2014-04-04 14:47:42	RPC	RPC/SRV1/CALLNATx	SQUARE	MCDRB02	IDL Tester	drb	4,865.600	Rsp 0007	RPC/SRV1/CALLNATx not registered
2014-04-04 14:47:12	RPC	RPC/SRV1/CALLNATx	SQUARE	MCDRB02	IDL Tester	drb	898.000	Rsp 0007	RPC/SRV1/CALLNATx not registered
2014-03-04 14:41:03	RPC	RPC/SRV1/CALLNAT	SQUARE	MCDRB02	IDL Tester	drb	454.000	Rc 0007	RPC/SRV1/CALLNAT not registered
2014-02-28 14:47:58	IMS Co...	x:12	CALC	daeexx23			9,096.800	20120353	java.net.UnknownHostException: a
2014-02-28 14:47:57	CJCS ECI	a:12	CALC	daeexx23			9,096.800	20120353	java.net.UnknownHostException: a
2014-02-28 14:47:51	RPC	RPC/SRV1/CALLNAT	SQUARE	MCDRB02	IDL Tester	drb	454.000	Rc 0007	RPC/SRV1/CALLNAT not registered
2014-02-28 14:47:47	RPC	RPC/SRV1/CALLNATx	SQUARE	MCDRB02	IDL Tester	drb	2,217.600	00070007	RPC/SRV1/CALLNATx not registered
2014-02-28 14:47:42	RPC	RPC/SRV1/CALLNATx	SQUARE	MCDRB02	IDL Tester	drb	865.600	00070007	RPC/SRV1/CALLNATx not registered

Transaction ID: **1000007**

Recording Time: **2014-02-28 14:47:47**

Response Time: **2,217.600 ms**

Scenario: **RPC**

Application: **RPC/SRV1/CALLNATx**

Address: **daeexx23:1972/RPC/SRV1/CALLNATx**

Program: **SQUARE**

Host (Client): **MCDRB02**

Application (C): **IDL Tester**

User (Client): **drb**

Error Code: **00070007**

Error Message:
RPC/SRV1/CALLNATx not registered

The upper table shows an overview of all error codes and messages. For each error code, you can see how often this error has occurred and its average, maximum and total response times. The columns for the average and maximum response times are colored according to the settings in the *Properties.csv* file, indicating good, average or poor performance.

The lower table shows the application monitoring KPIs for each transaction. These are the KPIs for the failed requests as described in [KPI Definitions for Application Monitoring](#).

In addition to the application monitoring KPIs, the table lists the **Transaction ID** which uniquely identifies each transaction. The transaction ID consists of the sequence number of the currently used data file in the *appmon.overview.v1.csv* file and (for the last six digits) the number of the record in the data file. By default, the transactions in the table are sorted by the transaction ID. You can change the sorting by clicking on the header of any column.

The **Response Time** column is colored according to the settings in the *Properties.csv* file, indicating good, average or poor performance.

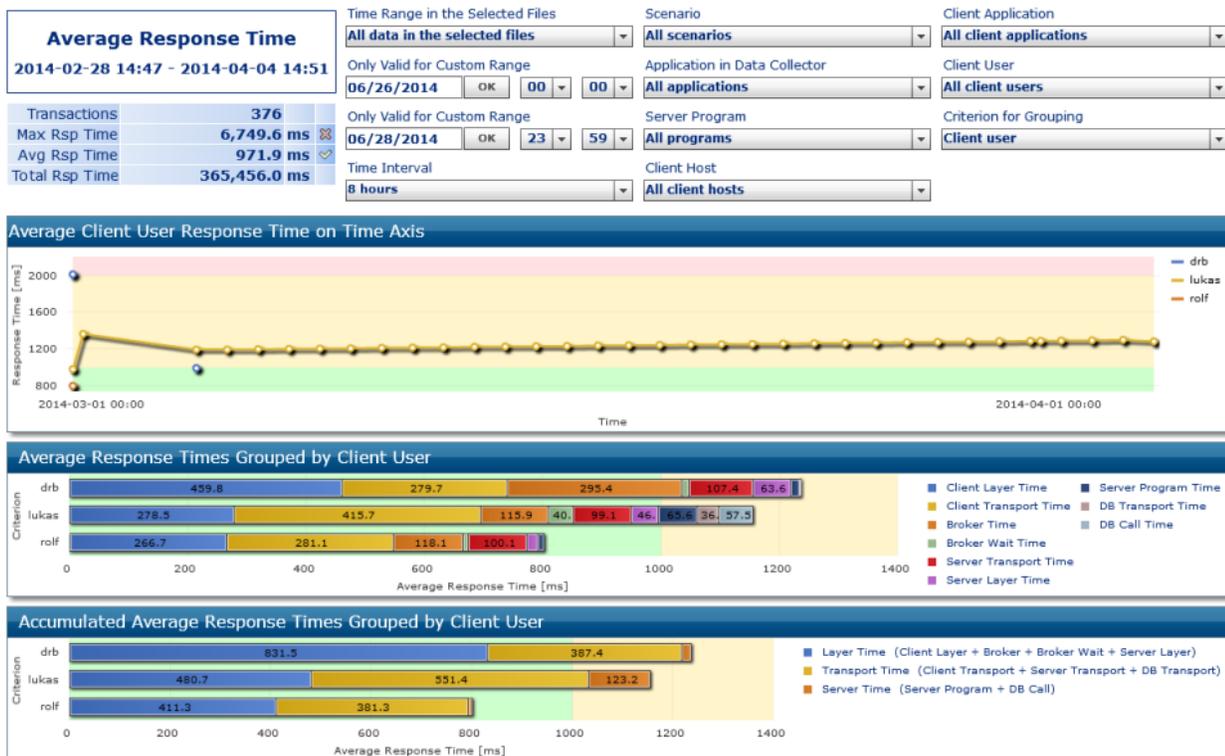
When you select (click) a line in the upper table, only the transactions with the selected error code and message are shown in the lower table.

When you select (click) a transaction in the lower table, all available column values are shown at the bottom of the tab. The bar chart below the table shows the response time of the currently selected transaction and compares it with the minimum, average and maximum response times of all transactions in the table. By default (if no transaction is selected), the chart shows the average response time and the background is colored according to the settings in the *Properties.csv* file, indicating good, average or poor performance.

The summarized data at the top of the tab shows the number of error transactions and the average, maximum and total response times for all error transactions in the selected time range for the selected components. For the average and maximum response times, icons indicate good, average or poor performance according to the settings in the *Properties.csv* file.

Average Response Time

The **Average** tab shows the average response times of the selected successful transactions grouped by scenario, application, program, client application, client user or client host according to the setting of the **Criterion for Grouping** selection box. By default, all data is summarized into one group.



The average response time is an indicator for the performance of your distributed application. The lower the average response time, the better the performance. Good, average or poor performance is indicated by the colored background according to the settings in the *Properties.csv* file.

The line chart at the top shows the history of the average response time. There is one line for each instance of the selected grouping criterion. The data in the line chart is aggregated according to the selected aggregated time interval.

The bar chart in the middle shows the distribution of single measured times. There is one bar for each instance of the selected grouping criterion.

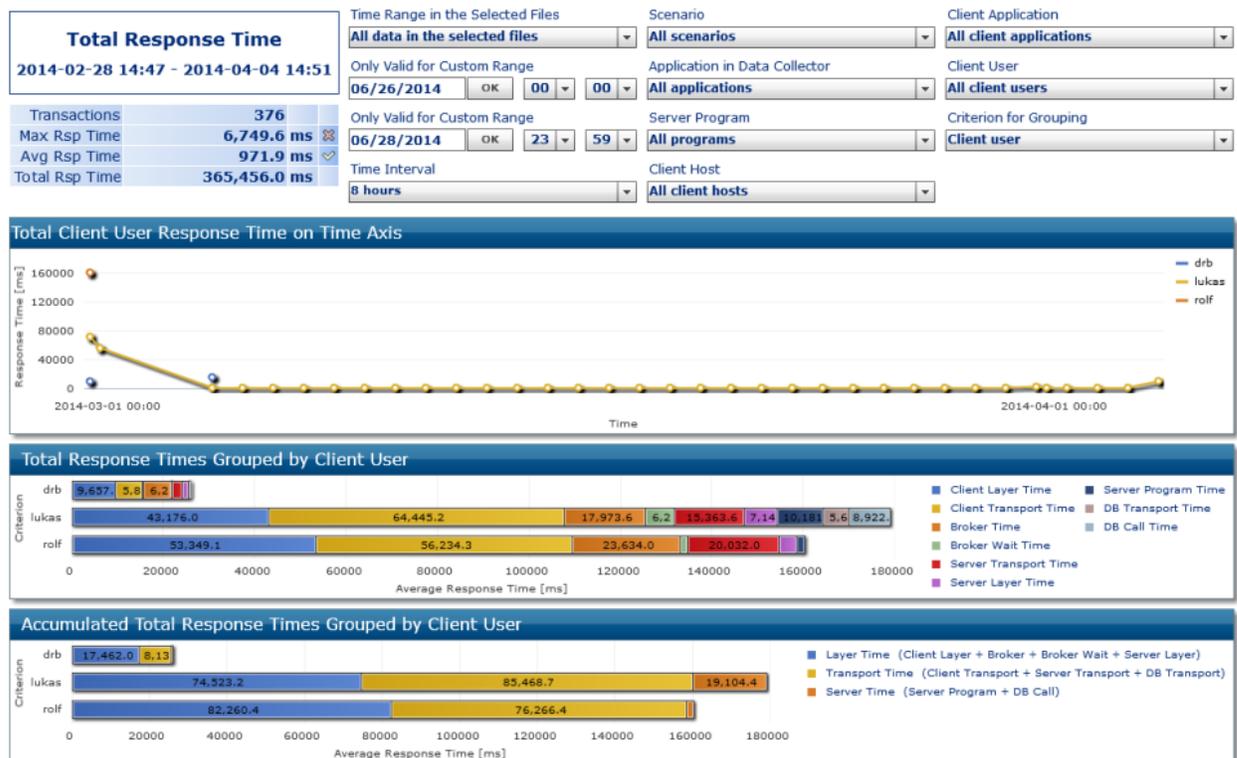
The bar chart at the bottom shows the measured times in three groups: layer time, transport time and server time. There is one bar for each instance of the selected grouping criterion.

By default, both bar charts show the average time of all selected transactions. When you click on a point in the line chart, the bar charts show only the response times of the corresponding point of time.

The summarized data at the top of the tab shows the number of successful transactions and the average, maximum and total response times for all successful transactions in the selected time range for the selected components. For the average and maximum response times, icons indicate good, average or poor performance, according to the settings in the *Properties.csv* file.

Total Response Time

The **Total** tab shows the total response times of the selected successful transactions grouped by scenario, application, program, client application, client user or client host according to the setting of the **Criterion for Grouping** selection box. By default, all data is summarized into one group.



The total response time is an indicator for the system load consumed by your distributed application. Even an application with a good performance may have a high system load if it is called very often.

The line chart at the top shows the history of the total response time. There is one line for each instance of the selected grouping criterion. The data in the line chart is aggregated according to the selected aggregated time interval.

The bar chart in the middle shows the distribution of single measured times. There is one bar for each instance of the selected grouping criterion.

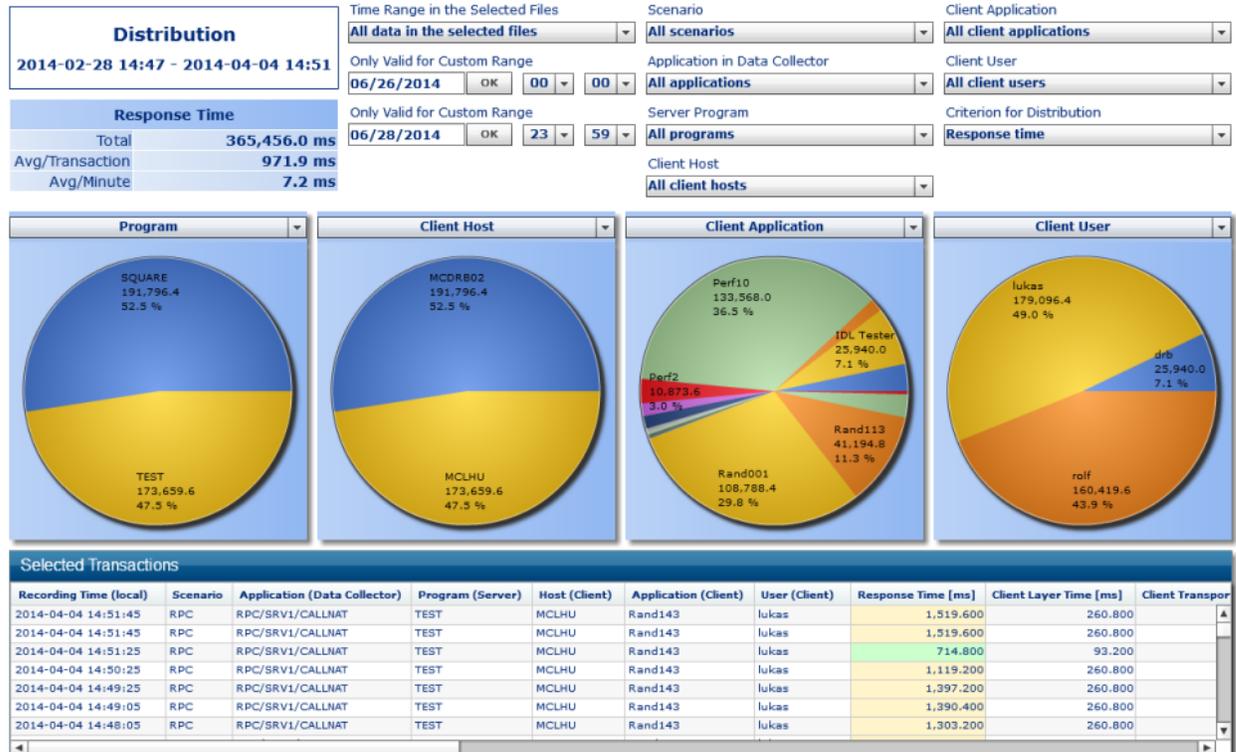
The bar chart at the bottom shows the measured times in three groups: layer time, transport time and server time. There is one bar for each instance of the selected grouping criterion.

By default, both bar charts show the total time of all selected transactions. When you click on a point in the line chart, the bar charts show only the response times of the corresponding point of time.

The summarized data at the top of the tab shows the number of successful transactions and the average, maximum and total response times for all successful transactions in the selected time range for the selected components. For the average and maximum response times, icons indicate good, average or poor performance, according to the settings in the *Properties.csv* file.

Distribution

The **Distribution** tab shows four pie charts. Each pie chart displays the distribution of the selected KPI over the instances of a selected component. Only successful transactions are evaluated in this tab.



You can select the KPIs from the following areas from the **Criterion for Distribution** selection box:

- Transactions
- Response time
- Layer time
 - Client layer time *
 - Broker time *
 - Broker wait time *
 - Server layer time *
- Transport time
 - Client transport time *
 - Server transport time *
 - DB transport time *
- Server time
 - Server program time *
 - DB call time *
- DB calls

The layer, transport and server times are summaries of other KPIs. These KPIs are marked by an asterisk (*) in the above list. In the selection box of the MashApp, these KPIs are indicated by a leading ">".

Each pie chart shows the distribution for the component which is currently selected in the selection box directly above the pie chart. For each pie chart, you can select one of the following components:

- Scenario
- Application
- Program
- Client Host
- Client Application
- Client User

The selections in the master view at the top of the page can be used as pre-selections, for example, if only the values of a specific application or user are to be evaluated.

When you select (click) a section in a pie chart, the corresponding component instance will be used as a filter for the subsequent pie charts. For example, when you click the section for the SQUARE program (see the above image), the other pie charts to the right will show only those client hosts, client applications and client users which have called the SQUARE program.

The table at the bottom lists the successful transactions filtered by the selections in the pie charts. The columns in the table are the same as in the [Successful Transactions](#) function.

The summarized data at the top of the tab shows the following values for the currently selected distribution KPI: the total, the average per transaction and the average per minute.

High Values

The **High Values** tab shows for each component the number of component instances, the name of the most-used instance and the percentage of the transactions referring the most-used instance.

High Values		Time Range in the Selected Files		Scenario	Client Application
2014-02-28 14:47 - 2014-04-04 14:51		All data in the selected files		All scenarios	All client applications
Number of Transactions		Only Valid for Custom Range		Application in Data Collector	Client User
Success	376	06/26/2014	00 00	All applications	All client users
Error	8	Only Valid for Custom Range		Server Program	Transaction Type
Total	384	06/28/2014	23 59	All programs	All transaction types
				Client Host	
				All client hosts	

Component	Instances	Most Used Instance	Percentage
Application	4	RPC/SRV1/CALLNAT	98.1 %
Address	5	daeexx23/RPC/SRV1/CALLNAT	97.9 %
Program	3	SQUARE	59.3 %
Client application	17	Perf10	49.7 %
Client host	3	MCDRB02	59.3 %
Client user	4	rolf	52.0 %

KPI	Average per Transaction	Maximum	Total
Response time [ms]	1,024.4	9,096.8	393,404.3
Number of DB calls	1,004.4	345,671	385,699
Request length	426.5	547	163,779

The following information is shown for some KPIs: the average value per transaction, the maximum value and the total. Note that database calls are only counted for calls against a Natural RPC server on a mainframe, whereas the **Average per Transaction** refers to all transactions.

If desired, the values can be restricted to successful transactions or error transactions only. By default, all transaction types are evaluated.

The summarized data at the top of the tab shows the total number of successful transactions and error transactions and the total of both for the selected time range and components.

Aggregation

The **Aggregation** tab shows summarized information for all instances of the selected aggregation criterion.

Application Monitoring Overview

The **Overview** tab provides a short description of the functions, charts and selections of the Application Monitoring MashApp.

Application Monitoring Overview			
Application monitoring enables you to monitor the response times and error situations in your distributed applications. The EntireX Application Monitoring Data Collector collects the response time data of each involved software component of selected synchronous EntireX RPC, CICS ECI or IMS Connect services and stores it in CSV (comma-separated values) files. These files are processed by the Application Monitoring MashApp. The MashApp refreshes every minute so that it always displays the most recent data.			
Functions of Application Monitoring			
Tab	Function	Description	Special Selection
Response Time	Response Time Monitor	History of general performance, response time peaks and system load	Criterion for Aggregation
Transactions	Transaction Monitor	History of general service usage	
Successful	Successful Transactions	Response time analysis for successful transactions	
Error	Error Transactions	Response time analysis for error transactions	
Average	Average Response Time	Performance analysis	Criterion for Grouping
Criterion for Aggregation			
Value	Description		
Average	History of general performance		
Maximum	History of general response time peaks		
Total	History of general system load		
Master View Elements of Function Response Time Monitor			
Name	Element Type	Description	
Function	Text (left)	The Application Monitoring MashApp function.	
Time Range	Text (left)	The currently selected time range.	
Summarized data	Table	The average, maximum or total response time for the selected successful transactions and error transactions. For the average and maximum response times, icons indicate good, average or poor performance.	
Time Range in the Selected Files	Selection box	Select the time range for the current function. The "Last ..." time ranges always use the last entry in the selected data files as the basis. The time range "All data in the selected files" refers to all data in the data files that are currently selected on the Data Files tab. When you select "Custom range", you can specify the desired start and end times in the following two lines.	
Only Valid for Custom Range / Custom Range From	Date input field and time selection boxes	These fields are only valid if the time range "Custom range" has been selected. Specify the desired start time (date, hour and minute) for the time range. The custom range is not restricted to the data of the currently selected data files.	
Only Valid for Custom Range / Custom Range To	Date input field and time selection boxes	These fields are only valid if the time range "Custom range" has been selected. Specify the desired end time (date, hour and minute) for the time range. The custom range is not restricted to the data of the currently selected data files.	
Charts of Function Response Time Monitor			
Name	Chart Type	Description	
Response Time of Successful Transactions	Bar chart (upper)	History of the average, maximum or total application response time for successful transactions. Good, average or poor performance is indicated by a green, yellow or red colored background, respectively.	
Response Time of Error Transactions	Bar chart (lower)	History of the average, maximum or total application response time for error transactions. Good, average or poor performance is indicated by a green, yellow or red colored background, respectively.	

When you click an entry in the **Tab** or **Value** column, the corresponding page of the MashApp is shown with all selections set to their initial default values (including the data file selection).