Chapter 5

AppMetrics Reports

An application monitor generates large amounts of data and stores them in log file format. To view this data in a manageable and meaningful way, AppMetrics Reports can display the data in various Microsoft® Excel® charts and tables. You can use these charts and tables to evaluate an application's performance during a specified time period.

Any application monitor can be the source for reports, but the monitor must first be configured to upload data from its logs to the database for AppMetrics Reports. When a monitor is configured this way, AppMetrics creates a database named after the monitor. It will then upload the data from the logs into the database at a specified interval. Once data is uploaded from the logs to the database, you can create and view a set of reports in AppMetrics Reports based on the data.

Requirements

Software

- Microsoft SQL Server® 2005 or SQL Server 2008
- Microsoft Excel 2000 or higher

Installation

The AppMetrics Reports modules are automatically copied to the machine and configured during installation of the AppMetrics manager software.

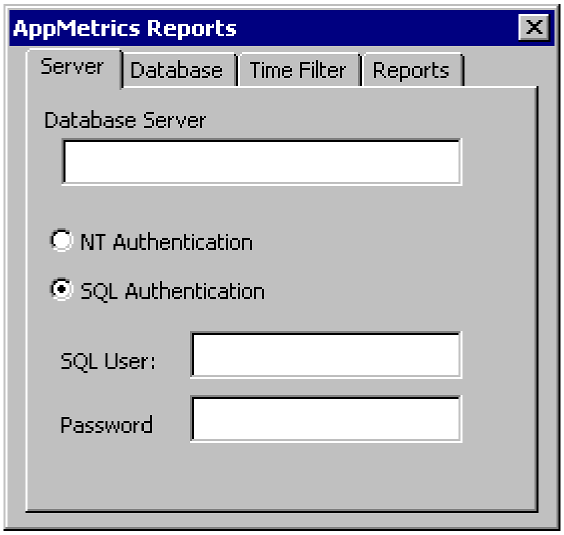
Configuration

Prior to viewing reports for a monitor's data, you must first configure the monitor to upload the data from its logs to the database. Refer to the Logging Options View section for each monitor type in Chapter 3.

Opening AppMetrics Reports

1. Log in to the machine with **AppMetrics Administrators** privileges for the Manager machine.
2. From the **Start** menu, select **Programs** → **Xtremesoft** → **AppMetrics for Transactions** → **AppMetrics Reports**.
3. If you receive a prompt about macros, click the option to enable macros.

   Within the Excel window, the AppMetrics Reports dialog opens:



Database Server

○ NT Authentication

● SQL Authentication

SQL User:

Password

Figure 5-1 AppMetrics Reports Dialog

Creating AppMetrics Reports for Monitors

The AppMetrics Reports dialog (Figure 5-1) enables you to create a specific set of reports. It contains the tabs where you choose the machine name, dataset (monitor name), time period, and report set. Once you enter the information in this dialog, you can then create the reports.

Note: You should complete each dialog tab in consecutive order, starting with the Server tab before proceeding to the next. The following four subsections describe each tab.

The AppMetrics Reports dialog appears when you start AppMetrics Reports. However, the dialog disappears after you create a report. To reopen the dialog at any time, right-click any non-chart cell, and then choose "AppMetrics" in the popup menu.

Server Tab

The Server tab (as shown in Figure 5-1) is used to specify the machine name and/or SQL instance name where the AppMetrics data is located. It is also used to specify the type of authentication used to connect to the SQL instance, and a SQL account name if using SQL authentication.

The **Database Server** field is left blank if the local machine is running a default SQL instance and is the machine used by the AppMetrics manager to store data.

Otherwise, a machine name and/or a SQL instance name must be specified as follows;

- Running reports against a default SQL instance on the local machine:
  - Leave the Database Server field blank
- Running reports against a named SQL instance on the local machine:
  - Specify the instance name in the form MACHINENAME\INSTANCENAME
- Running reports against a default SQL instance on a remote machine:
  - Specify the remote machine name only
- Running reports against a named SQL instance on a remote machine:
  - Specify the instance name in the form MACHINENAME\INSTANCENAME

In the **Authentication** area of the dialog, specify how you want to connect to the Diagnostic monitor database. Your specified account must have the PUBLIC database role in the Diagnostic monitor database. Through this role, the account gains the required permissions to run the stored procedures in the database.

To specify an account, perform the steps in only one of the following bulleted items:

- To use the current Interactive User account, select NT Authentication.

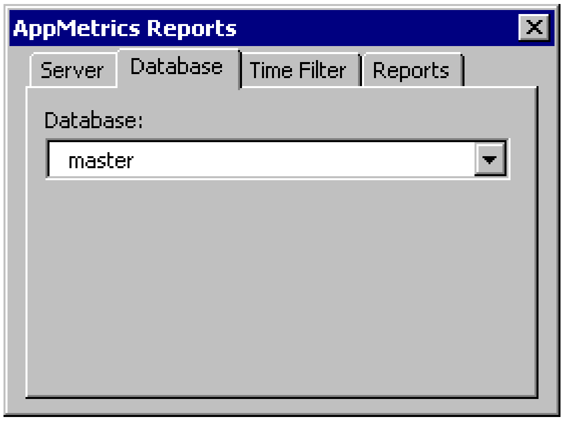
- Or -

- To specify a SQL Server login account, select SQL Authentication, type the login account name in the SQL User box, and then type its Password.

Notes

- If you enter a name and then click another tab, AppMetrics Reports will try to connect to the specified machine. While this occurs, it may take several seconds before activity in the dialog resumes.
- If AppMetrics is not installed on the machine, a prompt will notify you about the situation.
- If AppMetrics is installed on the machine, but none of its monitors are configured to upload data to the database, then a prompt will notify you about the situation.

Database Tab



Database: master

Figure 5-2 Database Tab

After specifying the machine name, you can then select the database for the Diagnostic monitor. The database typically has the same name as the monitor. Based on the authentication level of your login account, the drop-down list for the Database box may or may not show the existing databases on the server:

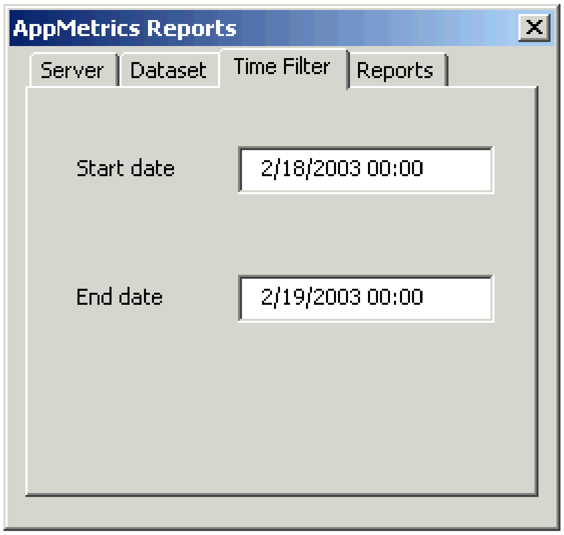
- If the drop-down list has no databases, then type the name of the Diagnostics monitor database.

- Or -

- If the list has the databases, click the drop-down arrow, and then select the database for the Diagnostics monitor.

Note: For databases targeting COM+ Production Manager monitors, the reports may display no data for time periods when detail throttling was in effect. See the "Detail Configuration View" section in Chapter 3 for more information.

Time Filter Tab



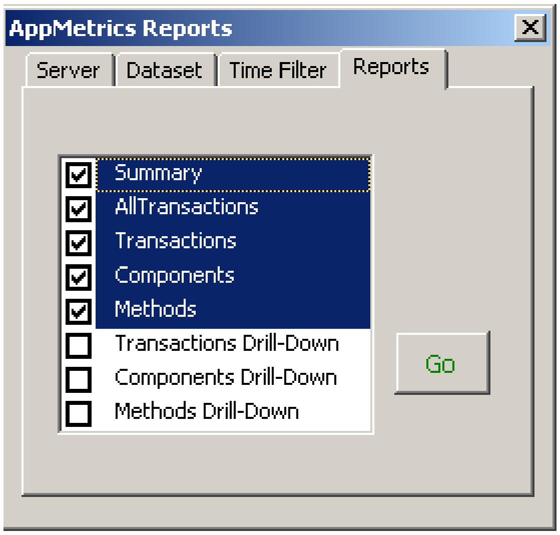
Start date 2/18/2003 00:00

End date 2/19/2003 00:00

Figure 5-3 Time Filter Tab

After choosing both the machine name and dataset, you can select the time period of the data to be viewed. The tab provides dates and times that are available for the chosen monitor in the Dataset tab.

Reports Tab



☑ Summary
☑ AllTransactions
☑ Transactions
☑ Components
☑ Methods
□ Transactions Drill-Down
□ Components Drill-Down
□ Methods Drill-Down

Figure 5-4 Reports Tab

After completing the information in the first three tabs, you can then select which reports to generate. The list of available reports in this tab depends on the monitor type chosen in the Dataset tab (Figure 5-2).

Each report results in its own worksheet in the Excel window. By default, all the reports are selected. If desired, you can deselect the ones you do not want to generate.

Refer to Production Monitor Reports on Page 5-6, Diagnostics Monitor Reports on Page 5-8 for a description of each report.

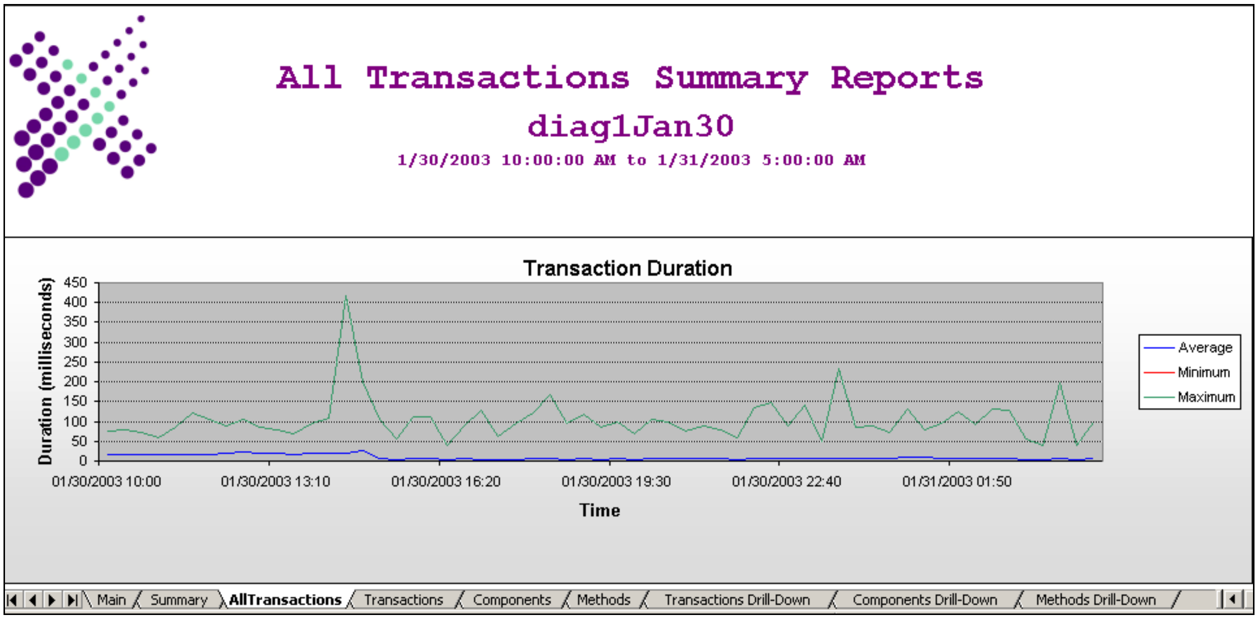
Generating Reports

After you complete the information in all four tabs, click **Go** in the Reports tab (Figure 5-4). AppMetrics Reports will gather the data based on your specified criteria and then create the reports. It may take several seconds to several minutes before the Excel window displays the reports. The length of the delay will depend on the amount of data to be loaded. In the case of monitors on non-local machines, the delay may also depend on network activity.

Note: When loading data generated by a monitor, you may receive warning prompts indicating that data was not found. This may occur if you selected a report in the Reports tab that corresponds to an ignored item. For example, if you choose Components in the Reports tab, but the monitor selected in the Dataset tab is configured to ignore components, then when you generate the report, AppMetrics informs you that no data could be found for components. When these messages appear, they can be ignored, and no report will be generated for the missing items.

For more information about what items a monitor is configured to observe, refer to the monitor configuration sections in Chapter 3. For Production monitors, see the sections for either the “Detail Configuration Panel” or “Applications Configuration Panel”. For Diagnostic monitors, refer to the “Applications Configuration Panel” section.

Reports



All Transactions Summary Reports

diag1Jan30

1/30/2003 10:00:00 AM to 1/31/2003 5:00:00 AM

Figure 5-5 AppMetrics Reports - Components

The Excel file (Figure 5-5) contains one tabbed worksheet for each report. More specifically, each worksheet corresponds to a report that was selected in the Reports tab (Figure 5-4).

The worksheets contain charts for the applications, transactions, components and methods that have been monitored. Some worksheets also provide tables that list the measured items. For example, the Components report contains a table with all the individual component types measured by the monitor. Additionally, the worksheets that include a table also provide a floating dialog, which enables you to choose the individual item to show in the graphs.

In most of the worksheets, you will have to scroll down to see the additional charts and tables.

The remainder of this chapter describes the available reports for both Production and Diagnostic monitors.

Production Monitor Reports

Summary Report

The Summary Report shows which items consume the most resources on the system. In particular, it displays charts of the top 10 transactions and components. In these charts, AppMetrics Reports calculates the number of starts multiplied by the average duration.

Applications Report

The Applications Report for Production monitors has four charts:

- Application Percent CPU – Shows the portion of the CPU used by the application during each interval. It can help determine if the currently charted application is using an excessive amount of CPU.
- Application Memory – Shows how much memory was being used by the application during each successive interval. A rising chart may indicate the application has a memory leak.

- Application Page faults per Second – Shows the rate of page faults. An excessive rate may result from low amounts of RAM on the server.
- Application Threads – Shows how many threads were used by the application during each interval.

Note If you're running .NET, the values tracked by these charts are for the sum of all instances of the application.

The charts display some of the same data that is available in the Applications runtime panel for Production monitors.

Beneath the last chart, the report also includes a table that lists the individual applications measured by the monitor. For each individual application measured during the time period of the report, the table provides the average, minimum and maximum values. Note that any value below one millisecond is displayed as a zero value in the table.

The AppMetrics - Application floating dialog enables you to choose which individual application to show in the charts. Click the drop-down arrow, select the application, and then click the Refresh Charts button.

Transactions Report

The Transactions Report for Production monitors has four charts:

- Active Transactions
- Transaction Rate (Started - Completed - Aborted)
- Transaction Duration
- DTC Transaction time / User Transaction time

The charts display the same data that is available in the Transactions runtime panel for Production monitors. You can use the charts to determine which transaction types are aborting or taking too long to complete.

Beneath the last chart, the report also contains a table that lists the individual transaction types measured by the monitor. For each individual transaction type measured during the time period of the report, the table provides the average, minimum, and maximum values. Note that any value below one millisecond is displayed as a zero value in both the table and the charts.

The AppMetrics - Transactions floating dialog enables you to choose which individual transaction type to show in the charts.

All Transactions Report

The All Transaction Report for Production monitors contains the following charts:

- Active Transactions
- Transactions (Starts - Completes - Aborts) charts

These are the same as the first two charts in the Transactions report, but the data within them are aggregates of all the transactions.

Components Report

The Components Report for a Production monitor has three charts:

- Active Component Instances
- Components (Starts - Completes - Aborts)
- Component Duration

The charts display the same data that is available in the Components runtime panel for Production monitors. You can use the charts to determine which component types are aborting or taking too long to complete. A large number of active components can show where resources are being consumed.

Beneath the last chart, the report also contains a table that lists the individual component types measured by the monitor. For each individual component type measured during the time period of the report, the table provides the average, minimum, and maximum values. Note that any value below one millisecond is displayed as a zero value in both the table and the charts.

The AppMetrics - Components floating dialog enables you to choose which individual component type to show in the charts.

Diagnostics Monitor Reports

Summary Report

The Summary Report for Diagnostics monitors shows which items consume the most resources on the system. In particular, it displays charts for the top 10 transactions, components, and if available, methods. In these charts, AppMetrics Reports uses the total duration to calculate the top events.

Transactions Report

The Transactions Report for Diagnostics monitors has two charts:

- Transaction Duration
- Transaction Completions

You can use the charts to determine if transaction types are taking too long to complete.

Beneath the last chart, the report also contains a table that lists the individual transaction types measured by the monitor. For each transaction type measured during the time period of the report, the table provides the average, minimum, and maximum values for duration and completions.

The AppMetrics - Transactions floating dialog enables you to choose which individual transaction type to show in the charts.

All Transactions Report

The All Transactions Report contains the same data as the Transactions Report for Diagnostics monitors, but it shows an aggregate of the data for all the transactions instead of data for individual transactions.

Components Report

The Components Report for a Diagnostics monitor has two charts:

- Component Duration
- Component Completions

You can use the charts to determine if the currently charted component is taking too long to complete.

Beneath the last chart, the report also contains a table that lists the individual component types measured by the monitor. For each individual component monitored during the time period of the report, the table provides the average, minimum, and maximum values for duration and completions.

The AppMetrics - Components floating dialog enables you to choose which individual component type to show in the charts.

Methods Report

The Methods Report, if available for any particular Diagnostics monitor, contains two charts:

- Method Duration - shows the average, minimum, and maximum lengths of time for the calls of the currently charted method.
- Method Calls - shows how many times the currently charted method was called during each interval.

The charts can provide an indication of which methods types are running slowly.

Beneath the last chart, the report also contains a table that lists the individual method types measured by the monitor. For each individual method type that was called and monitored during the time period of the report, the table provides the average minimum, and maximum values for duration and completions.

The AppMetrics - Methods floating dialog enables you to choose which individual method type to show in the charts.

Transactions Drill-Down Report

The Transactions Drill-Down Report shows user transactions, which consist of their root objects and the method(s) that are called within them. It sorts information by transaction, component, and method start times.

Use the Transactions Drill-Down Report to discover what components take part in a given transaction. You can also use the report to find out what methods in these components are called during the transaction. This report is useful when many components are involved in a transaction. It can show you how long components are active before they are first called or if they are ever called at all during the transaction.

The Transactions Drill-Down Report sorts the items first by Transaction start times, then by Component start times, and finally by Method start times. Understanding this sort order is key to following the report.

Note: The Transactions Drill-Down Report does not tell you the logical chain of method calls for a transaction. Instead, you can see the logical chain of method calls more clearly in the Methods report, which is described later.

The following table describes each column in the Transactions Drill-Down Report:

Column	Description
Transaction	The name of the user transaction. This name consists of the component type that was called to start the transaction.
Component	The name of a component that became active within the transaction. • When a detected component resides in a different package/application from the currently chosen package/application, the report adds the package/application name to the beginning of the component name. For example, **Package2 : Proj2.ComponentC** implies that *Proj2.ComponentC* resides in *Package2*. • If a server package calls a component in a library package, the drill-down report refers to the library component not by the name of its library package, but by the name of the server package. For example, suppose you have the following two components: o *Proj2.ComponentC* in server package *Package2* o *Proj3.ComponentD* in library package *Package3* If the former calls the latter, then the report lists the latter as **Package2 : Proj3.ComponentD**.

Column	Description
Method	The name of the method that was called in the component. This name consists mainly of the component and the method. For example, **Proj2.ComponentC->Method4**. **Notes** • This data is only available if the Method Call Detail setting was on during monitoring in the Diagnostics monitor. • When the component for a method resides in a different package/application from the currently chosen package/application, the report adds the package/application name to the beginning of the method name. For example, **Package2 : Proj2.ComponentC->Method1** implies that *Proj2.ComponentC* resides in *Package2*. • If a server package calls a method in a library package, the drill-down report refers to the method not by the name of its library package, but by the name of the server package. For example, suppose you have the following two methods: o *Proj2.ComponentC->Method4* in server package *Package2* o *Proj3.ComponentD->Method5* in library package *Package3* If the former calls the latter, then the report lists the latter as **Package2 : Proj3.ComponentD->Method5**.
StartTime	The beginning date and time for the transaction, component, or method.
EndTime	The completion date and time for the transaction, component, or method.
RelativeStart (ms)	The beginning time of the component or method in relation to the start of the transaction. The RelativeStart time for the transaction itself is zero. This is more useful than the StartTime because it is more granular.
RelativeEnd (ms)	The completion time for the transaction, component, or method in relation to the start of the transaction. This is more useful than the EndTime because it is more granular.
Duration (ms)	The lifespan of the transaction, component, or method in milliseconds.
Error	The error code returned by the method.

Components Drill-Down Report

The Components Drill-Down Report shows calls into the components of monitored packages/applications. It sorts information by component and method start times.

Use the Components Drill-Down Report when you need to view method calls in relation to component instances of a particular package/application.

Although the report shows method calls on the component instance, this doesn't preclude these calls from calling other methods on other components. Nonetheless, this report excludes details about these subordinate calls. If you need such a view, use the Methods Drill-Down Report instead.

The Components Drill-Down Report sorts information first by Component start times, then by Method start times. Understanding this sort order is key to following the report.

The following table describes each column in the Components Drill-Down Report:

Column	Description
Component	The name of a component that became active during monitoring within the chosen package/application.
Method	The name of a called method in the component. **Note:** This data is only available if the Method Call Detail setting was on during monitoring in the Diagnostics monitor.
StartTime	The beginning date and time for the component or method.
EndTime	The completion date and time for the component or method.
RelativeStart (ms)	The beginning time of the method in relation to the start of the component. The RelativeStart time for the component itself is zero. This is more useful than the StartTime because it is more granular.
RelativeEnd (ms)	The completion time for the component or method in relation to the start of the component. This is more useful than the EndTime because it is more granular.
Duration (ms)	The lifespan of the component or method in milliseconds.
Error	The error code returned by the method.

Methods Drill-Down Report

The Methods Drill-Down Report shows the method calls for the transactions in the monitored packages/applications. It sorts information by transaction and method start times.

You can use the Methods Drill-Down Report to see the logical chain of method call activity in the chosen package/application. The report also lets you see the durations for individual method calls. In addition, the report includes information about cross-package calls.

In the report, you can use the Relative Start and Relative End times to find out which calls are subordinate to one another. In particular:

- If a call has a Relative Start time that is before the Relative End time of the prior call, then this call is subordinate to the prior call. For example, if a method call named B has a Relative Start time of 400, but the preceding method call named A has a Relative End time of 600, then the method call B is subordinate to the method call A.
- The report includes the duration of a subordinate call in the duration of the parent call.

The Methods Drill-Down Report sorts the items first by Transaction start times, and then by Method start times. Understanding this sort order is key to following the report.

The following table describes each column in the Methods Drill-Down Report:

Column	**Description**
Transaction	The name of the component type that was called to start the transaction.
Method	The name of the method in the component that was called. This name consists mainly of the component and the method. For example, **Proj2.ComponentC->Method4**. **Notes** • This data is only available if the Method Call Detail setting was on during monitoring in the Diagnostics monitor. • When the component for a method resides in a different package/application from the currently chosen package/application, the report adds the package/application name to the beginning of the method name. For example, **Package2 : Proj2.ComponentC->Method1** implies that *Proj2.ComponentC* resides in *Package2*. • If a server package calls a method in a library package, the report refers to the method not by the name of its library package, but by the name of the server package. For example, suppose you have the following two methods: o *Proj2.ComponentC->Method4* in server package *Package2* o *Proj3.ComponentD->Method5* in library package *Package3* If the former calls the latter, then the report lists the latter as **Package2 : Proj3.ComponentD->Method5**.
StartTime	The beginning date and time for the transaction or method.
EndTime	The completion date and time for the transaction or method.

Column	Description
RelativeStart (ms)	The beginning time of the method in relation to the start of the transaction. The RelativeStart time for the transaction itself is zero. This is more useful than the StartTime because it is more granular.
RelativeEnd (ms)	The completion time for the transaction or method in relation to the start of the transaction. This is more useful than the EndTime because it is more granular.
Duration (ms)	The lifespan of the transaction or method in milliseconds.
Error	The error code returned by the method.