Sun™ Mainframe Administration Tool User's Guide

Release 1.0.0

Sun Microsystems, Inc.
www.sun.com

Part No. 817-0572-11
February 2004, Revision A

Submit comments about this document at: http://www.sun.com/hwdocs/feedback

Please
Recycle

Contents

Figures

Tables

Preface

This document describes the Sun™ Mainframe Administration Tool software (Sun MAT).

How This Book Is Organized

Chapter 1 provides an introduction to Sun MAT.

Chapter 2 describes how to install Sun MAT and the agent.

Chapter 3 describes how to get started with Sun MAT.

Chapter 4 describes the tool's user interface.

Chapter 5 describes the tool's windows and how to manipulate them.

Chapter 6 describes the charting features of Sun MAT, and explains how to create graphs and gauges.

Chapter 7 describes the Overview tab and its contents.

Chapter 8 describes the System tab, its subscreens, and datapoints.

Chapter 9 describes the Communications tab, its subscreens, and datapoints.

Chapter 10 describes the Resources tab, its subscreens, and datapoints.

Chapter 11 describes the Advanced tab, its subscreens, and datapoints.

Using UNIX Commands

This document does not contain information about basic UNIX® commands and procedures such as shutting down the system, booting the system, and configuring devices. See the following for this information:

- Software documentation that you received with your system
- Solaris™ operating environment documentation, which is at

  `http://docs.sun.com`

Shell Prompts

Shell	Prompt
C shell	*machine-name*%
C shell superuser	*machine-name*#
Bourne shell and Korn shell	$
Bourne shell and Korn shell superuser	#

Typographic Conventions

Typeface[1]	Meaning	Examples
`AaBbCc123`	The names of commands, files, and directories; on-screen computer output	Edit your `.login` file. Use `ls -a` to list all files. `% You have mail.`
`AaBbCc123`	What you type, when contrasted with on-screen computer output	`% su` `Password:`
AaBbCc123	Book titles, new words or terms, words to be emphasized. Replace command-line variables with real names or values.	Read Chapter 6 in the *User's Guide*. These are called *class* options. You *must* be superuser to do this. To delete a file, type `rm` *filename*.

1 The settings on your browser might differ from these settings.

Related Documentation

Product	Title	Part Number
Sun Mainframe Transaction Processing software	*Sun Mainframe Transaction Processing Software Administrator's Guide*	816-5327-10
	Sun Mainframe Transaction Processing Software Configuration Guide	816-5328-10
	Sun Mainframe Transaction Processing Software Developer's Guide	816-5329-10
	Sun Mainframe Transaction Processing Software Message Guide	816-5331-10
	Sun Mainframe Transaction Processing Software Reference Guide	816-5334-10
	Sun Mainframe Transaction Processing Software Release Notes	816-5335-11
	Sun Mainframe Transaction Processing Software Troubleshooting and Tuning Guide	816-5333-10

Product	Title	Part Number
Sun Mainframe Batch Manager software	*Sun Mainframe Batch Manager Software Configuration Guide*	816-5342-10
	Sun Mainframe Batch Manager Software Installation Guide	816-5343-10
	Sun Mainframe Batch Manager Software Message Guide	816-5345-10
	Sun Mainframe Batch Manager Software Migration Guide	816-5346-10
	Sun Mainframe Batch Manager Software Reference Guide	816-5347-10
	Sun Mainframe Batch Manager Software Release Notes	816-5348-11
	Sun Mainframe Batch Manager Software User's Guide	816-5349-10
IBM CICS	*CICS Application Programming Reference*	SC33-1170
	CICS Application Programming Primer	SC33-0674
	CICS Master Index	SC33-1074
	CICS Supplied Transactions	SC33-1686
	CICS System Programming Reference	SC33-1689
Micro Focus Server Express	Micro Focus Server Express documentation CD-ROM	*
Liant Open PL/I	*Liant Open PL/I User's Guide*	*
	Liant Open PL/I Language Reference Manual	*
	Liant CodeWatch Reference Manual	*
C	C Compiler documentation	*
C-ISAM	*C-ISAM Programmer's Manual*	*
	System Performance Tuning by Mike Loukides, published by O'Reilly & Associates as part of their "Nutshell" series	

* These manuals vary with the platform being used. Consult your vendors for the manuals that are applicable to your platform.

Accessing Sun Documentation

You can view, print, or purchase a broad selection of Sun documentation, including localized versions, at:

http://www.sun.com/documentation

Contacting Sun Technical Support

If you have technical questions about this product that are not answered in this document, go to:

http://www.sun.com/service/contacting

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. You can submit your comments by going to:

http://www.sun.com/hwdocs/feedback

Please include the title and part number of your document with your feedback:

Sun Mainframe Administration Tool User's Guide, part number 817-0572-11

CHAPTER 1

Introduction to the Sun Mainframe Administration Tool

The Sun MAT software is designed to provide remote administrative access to one or more Sun Mainframe Transaction Processing (Sun MTP) regions running on one or more systems distributed throughout a network. The tool consists of three components:

- Sun MAT, the graphical user interface (GUI)-based client used to administer regions
- Sun Mainframe Administration Agent (Sun MAA), which provides the interface between the GUI and the regions being administered
- The `unikixadmin` process, which executes under the control of Sun MTP, and provides the interface to the Sun MTP system information

Each system hosting one or more Sun MTP regions must also host a running instance of Sun MAA to provide access to those regions. Sun MAT can run anywhere in the network.

In a simple environment where one system has one region to be administered, the three components (Sun MAT, Sun MAA, and Sun MTP) are all co-located, as shown in the following figure:

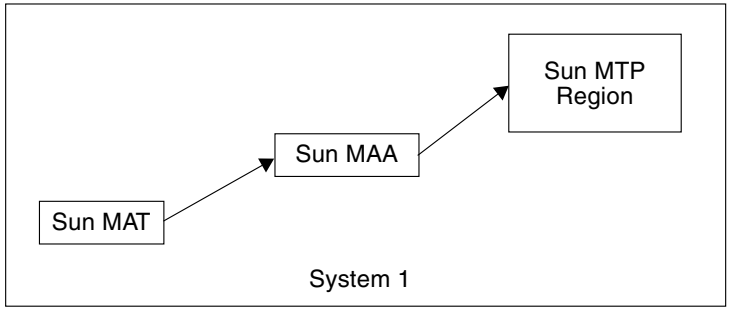

FIGURE 1-1 Single Region/Single System Environment

The following figure illustrates a more complex environment that has multiple regions running on multiple systems, all monitored from a single point.

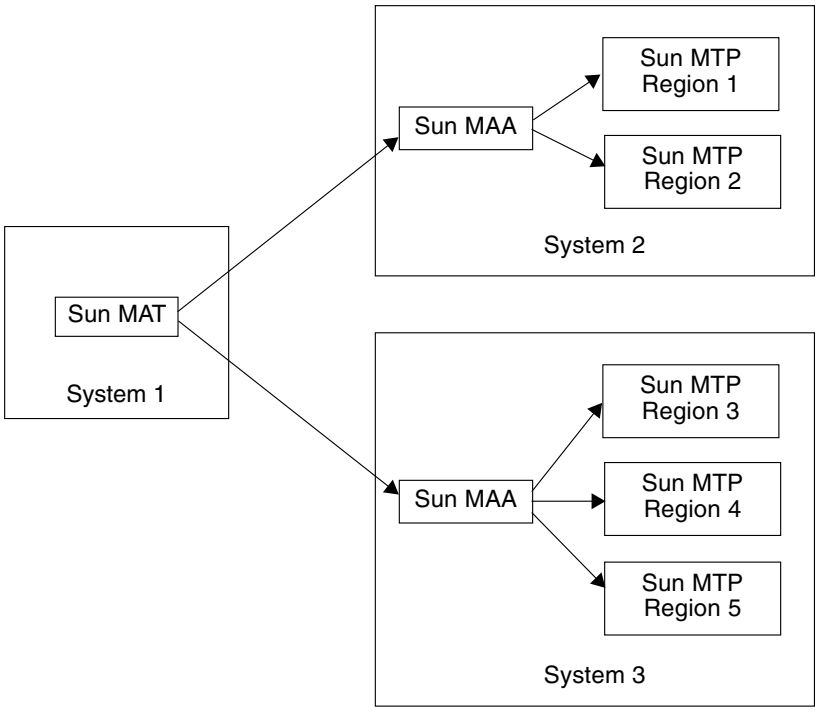

FIGURE 1-2 Multiple Region/Multiple System Environment

It is also possible for more than one administrator to run Sun MAT concurrently and access the same regions, as shown in FIGURE 1-3.

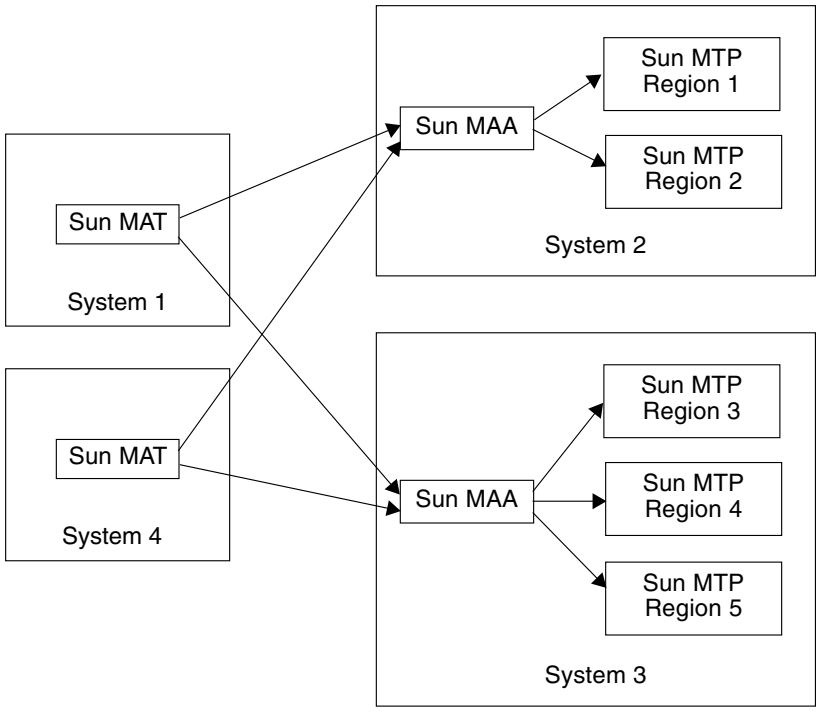

FIGURE 1-3 Multiple Regions Accessed by Multiple Sun MATs

CHAPTER 2

Installing Sun MAT

This chapter describes the Sun MAT software and explains how to install and start the Sun MAT components. It includes the following topics:

Prerequisites

Any Sun MTP regions that are to be remotely administered through Sun MAA must be at Release 8.0.0 or later.

Sun MAA is a Java application that requires Java 2 Standard Edition (J2SE) version 1.4. Before attempting to start Sun MAA, J2SE 1.4 must be installed on the system where you plan to run Sun MAA.

J2SE version1.4 must also be defined on the `$PATH` of the user that will start Sun MAA. To test whether J2SE 1.4 is properly defined, type the following commands. The output should be as shown:

```
$ which java
/usr/java1.4.0/bin/java
$ java -version
java version "1.4.0"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.4.0-b92)
Java HotSpot(TM) Client VM (build 1.4.0-b92, mixed mode)
```

The location of J2SE might be different on your system. Contact your system administrator to determine if and where J2SE 1.4 is installed. If J2SE 1.4 is not installed on your system, it can be downloaded from:

http://www.java.sun.com

Sun MAT is also a Java application, and can run anywhere that a J2SE 1.4 runtime is available.

Installation

This section describes how to install the Sun MAA and Sun MAT on the various platforms on which they run.

Installing Sun MAA

Sun MAA is distributed as a compressed file named `MAA1.0.0.zip`. It must be installed on each system on which there are Sun MTP regions that you want to administer.

Follow these guidelines when installing Sun MAA:

- Do not install Sun MAA as the root user. Install it as the same user that installed Sun MTP.
- You might find it convenient to install Sun MAA in a location analogous to that used for the installation of Sun MTP. For example, if Sun MTP is installed under the directory `/pkgs/mtp`, you could install Sun MAA under the directory `/pkgs/maa`. This is not a requirement, however.

▼ To Install Sun MAA

1. **Create the top-level installation directory:**

```
$ mkdir -p /pkgs/maa
```

2. **Change to the installation directory:**

```
$ cd /pkgs/maa
```

3. **Extract the archive. For example:**

```
$ unzip /cdrom/MAA1.0.0.zip .
```

Refer to the `unzip`(1) man page for information on command options.

The following directory structure is created:

```
/pkgs/maa/MAA1.0.0
/pkgs/maa/MAA1.0.0/bin
/pkgs/maa/MAA1.0.0/etc
/pkgs/maa/MAA1.0.0/lib
/pkgs/maa/MAA1.0.0/lib/dependencies
```

`MAA1.0.0` indicates the version of Sun MAA that you have installed.

Installing Sun MAT

Sun MAT provides an administrator with a graphical user interface (GUI) to remotely administer Sun MTP regions. It is provided in three distribution formats appropriate for different platforms. The distribution files are named as follows:

- `MAT1.0.0.tar` - For Solaris or other UNIX systems.
- `MAT1.0.0.exe` - For Microsoft Windows. This 32-bit InstallShield executable includes the J2SE 1.4.0 runtime software.
- `MAT1.0.0.zip` - Generic Java distribution, which is suitable for all other platforms.

Choose the installation file most appropriate for your platform.

Installing Sun MAT on Solaris or Other UNIX Systems

Follow these guidelines when installing Sun MAT:

- Do not install Sun MAT as the root user. If you are installing Sun MAT on the same system as Sun MAA and Sun MTP, you can use the same user ID as was used to install these products. However, this is not required.
- You might find it convenient to install Sun MAA in a location analogous to that used for the installation of Sun MTP and Sun MAA. For example, if Sun MTP is installed under the directory `/pkgs/mtp`, and Sun MAA under the directory `/pkgs/maa`, you could install Sun MAT under the directory `/pkgs/mat`. This is not a requirement, however.

▼ To Install Sun MAT on Solaris or Other UNIX System

1. **Create the top-level installation directory:**

```
$ mkdir -p /pkgs/mat
```

2. **Change to the installation directory:**

```
$ cd /pkgs/mat
```

3. **Extract the archive. For example:**

```
$ tar -xvof /cdrom/MAT1.0.0.tar
```

The following directory structure is created:

```
/pkgs/mat/MAT1.0.0
/pkgs/mat/MAT1.0.0/bin
/pkgs/mat/MAT1.0.0/doc
/pkgs/mat/MAT1.0.0/doc/images
/pkgs/mat/MAT1.0.0/lib
/pkgs/mat/MAT1.0.0/lib/dependencies
```

`MAT1.0.0` indicates the version of Sun MAT that you have installed.

Installing Sun MAT on Windows Systems

For 32-bit Microsoft Windows platforms, Sun MAT is supplied as a self-installing InstallShield executable. This executable contains an integrated J2SE 1.4.0 runtime.

▼ To Install Sun MAT on Windows Systems

- **Execute the `MAT1.0.0.exe` file and follow the on-screen instructions.**

Installing Sun MAT on Other Systems

Sun MAT can also be installed and run on other platforms supporting J2SE 1.4.

▼ To Install Sun MAT on Other Systems

1. **Create the top-level installation directory:**

```
$ mkdir -p /pkgs/mat
```

2. **Change to the installation directory:**

```
$ cd /pkgs/mat
```

3. **Extract the archive** `MAT1.0.0.zip` **to the directory you just created.**

   The following directory structure is created:

```
MAT1.0.0
MAT1.0.0/bin
MAT1.0.0/doc
MAT1.0.0/doc/images
MAT1.0.0/lib
MAT1.0.0/lib/dependencies
```

Note – The `bin` directory will be empty for this installation.

`MAT1.0.0` indicates the version of Sun MAT that you have installed.

Configuring and Starting Sun MAA

Before starting Sun MAA, you must decide on a port number on which the agent will listen, and you must choose the user ID that will start the agent.

Deciding on a Port Number

When Sun MAA starts, it listens for Java Remote Method Invocation (RMI) calls from instances of Sun MAT, by listening on a TCP/IP port. This port is 9980 by default. If you want to use a different port number, you can edit the `sunmaa` startup script, which is located in the `bin` directory of the Sun MAA installation. Instructions on editing the script are provided in the script itself.

Choosing an Appropriate User ID

Sun MAA should be started as the same UNIX user that was used to install it. This is usually the same UNIX user that is used to run each of the Sun MTP regions on the system.

In more complex environments, however, several regions can exist, each running as a different UNIX user. In this case, use the following guidelines to decide what UNIX user should run the agent:

- The agent must have read access to all of the Sun MAA product files. These are all files under the following directory:

  `/pkgs/maa/MAA1.0.0/`

- The agent must have read-write access to the following file:

  `/pkgs/maa/MAA1.0.0/etc/region_list.xml`

- The agent must have read access to files in the `$KIXSYS` directory for each region to which the agent is to provide access.
- The agent must have read access to files in the Sun MTP installation directory (`$UNIKIX`) for each version of Sun MTP for which the agent is to provide administrative access.

▼ To Start Sun MAA

- **Execute the `sunmaa` shell script, which is located in the Sun MAA `bin` directory:**

```
$ /pkgs/maa/MAA1.0.0/bin/sunmaa
```

Output similar to the following will be displayed on the screen:

```
MAA: Starting
MAAMaster: control file name is <etc/region_list.xml>
MAA: started RMI connector on port 9980
```

The `etc/region_list.xml` control file is the file in which Sun MAA stores information about each region for which it is responsible. Each region that you want to administer must be registered with the agent running on that system. See "Registering a Region With Sun MAA" on page 18.

Starting Sun MAT

This section describes how to start Sun MAT on the different platforms on which it is supported.

▼ To Start Sun MAT on Solaris Systems

1. **Verify that the J2SE 1.4.0 runtime is installed on the system and its location is defined in your `PATH` environment variable.**

   See "Prerequisites" on page 5.

2. **Execute the `sunmat` shell script, which is located in the Sun MAT `bin` directory:**

```
$ /pkgs/mat/MAT1.0.0/bin/sunmat
```

The Sun MAT GUI starts. See FIGURE 2-1.

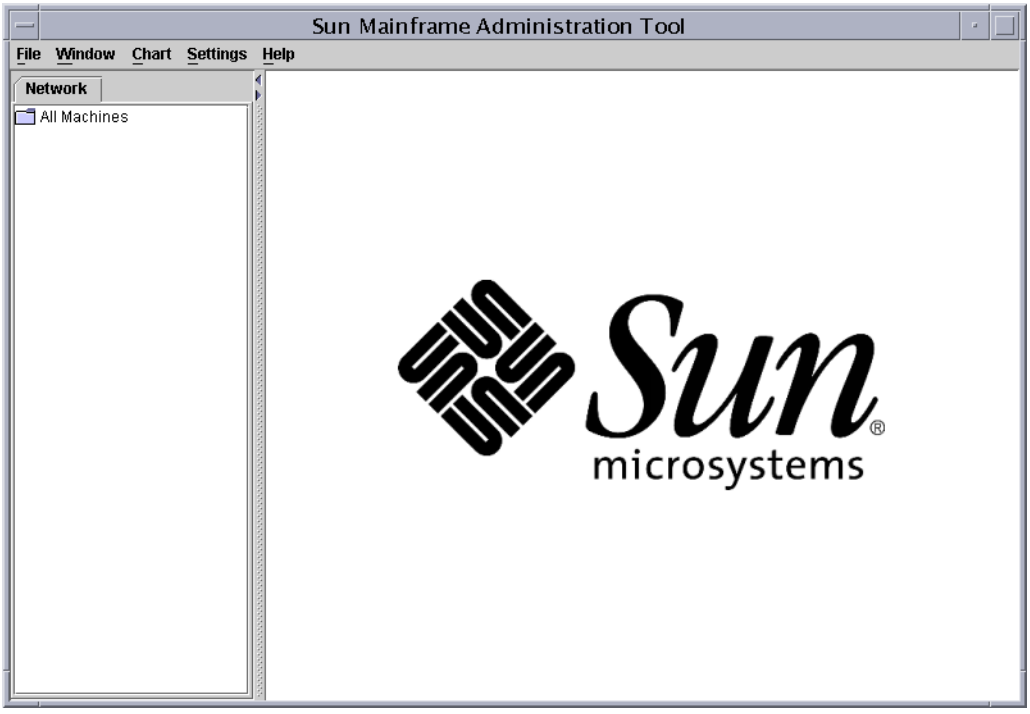

FIGURE 2-1 Sun MAT Startup Screen

▼ To Start Sun MAT on Windows

- **Use either method:**
  - On the Windows desktop, select Start → Programs → Sun MAT 1.0.0 → Administration Tool
  - Execute the `sunmat.exe` executable in the product's `bin` directory.

  The Sun MAT GUI starts. See FIGURE 2-1.

▼ To Start Sun MAT on Other Systems

1. **Verify that the J2SE 1.4.0 runtime is installed and available on your system.**
2. **Run the executable JAR file supplied in the `lib` directory of the Sun MAT installation. Depending on your system, use one of the following methods:**
   - Change to the `lib` directory and execute the following command:

```
$ java -jar sunmat.jar
```

   - Click on the `sunmat.jar` file using the GUI of your system.

   The Sun MAT GUI starts. See FIGURE 2-1.

CHAPTER 3

Getting Started With Sun MAT

Before you can start using the administration tool, there are several tasks to perform, which are described in the following sections:

- "Configuring Sun MAT to Access an Agent System" on page 16 describes how to add a system that hosts the regions you want to monitor.
- "Registering a Region With Sun MAA" on page 18 describes how to register regions that you want to monitor.
- "Setting Preferences" on page 22 describes the preferences that control the tool's startup and exit behaviors and the frequency with which the tool polls regions for information.

This chapter also contains instructions on removing regions and machines. See "Removing Regions and Machines" on page 21.

Accessing Sun MTP Regions

This section describes the procedures you must perform to access each of the regions you want to administer.

▼ To Access Sun MTP Regions

1. **Start Sun MAA.**

   See "Configuring and Starting Sun MAA" on page 10.

2. **Start Sun MAT.**

   See "Starting Sun MAT" on page 11.

3. **Configure Sun MAT to access each system on which an agent (Sun MAA) is running.**

   See "Configuring Sun MAT to Access an Agent System" on page 16.

4. **Register each region running on a system with the agent (Sun MAA) running on that system.**

   See "Registering a Region With Sun MAA" on page 18.

Configuring Sun MAT to Access an Agent System

The example in this section uses a simple configuration like the one shown in the following figure. The procedure also applies to complex configurations.

In this example, Sun MAT is running on the same system as Sun MAA, although it need not be. Sun MAA has been started listening on the default port of 9980.

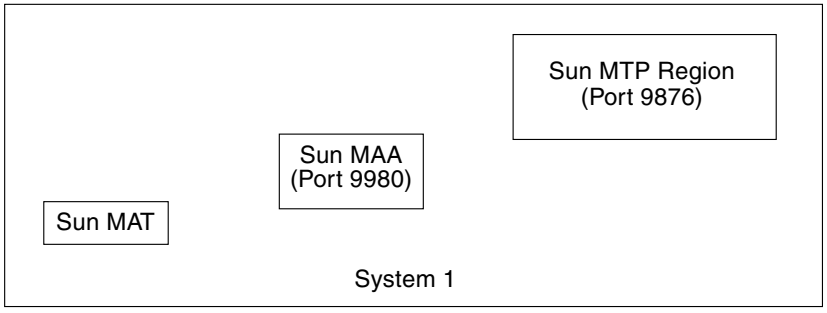

FIGURE 3-1 No Access From Sun MAT to a Region

▼ To Access an Agent System

1. **Make sure Sun MAA is started.**

   See "Configuring and Starting Sun MAA" on page 10

2. **Make sure Sun MAT is started and the GUI is displayed.**

   See "Starting Sun MAT" on page 11.

3. **On the GUI menu, choose File → Add Machine.**

   This option will make an agent known to Sun MAT. See FIGURE 3-2.

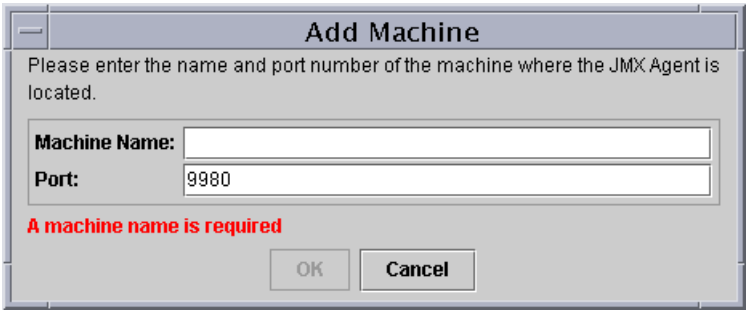

FIGURE 3-2 Adding a System

4. **In the dialog box:**

   a. **Type the name of the system on which the agent is running.**

   b. **Verify the port number on which the agent is listening.**

   c. **Click OK.**

      Sun MAT will contact the agent, and a machine entry will be displayed in the tree panel on the left side of the Sun MAT display. You might have to click the key icon next to the name to display the regions.

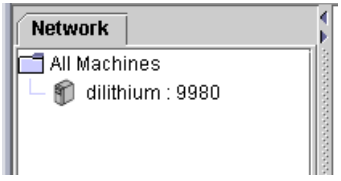

FIGURE 3-3 Machine Entry

Note – Sun MAT treats an instance of an agent and the system on which that agent is running as being synonymous, because there is typically only one agent instance on each system. Sun MAT, therefore, uses the terminology *Add Machine*.

Communication is now established between Sun MAT and Sun MAA, as shown in FIGURE 3-4.

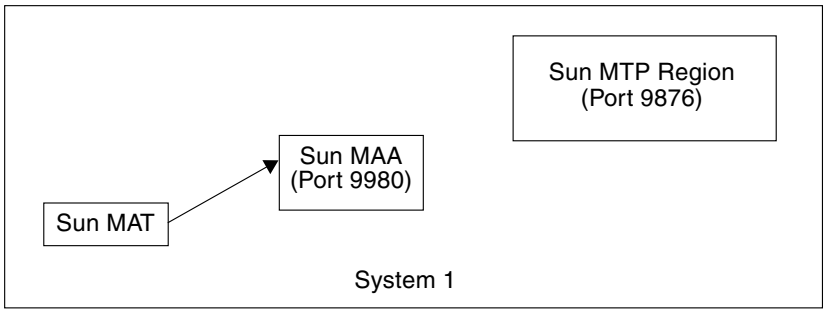

FIGURE 3-4 Sun MAT Access to Sun MAA

Registering a Region With Sun MAA

Before a region can be registered with an agent, the region must be started with the `unikixadmin` process enabled. The `unikixadmin` server is a daemon process that allows a region to be monitored by Sun MAT.

▼ To Enable the `unikixadmin` Server

1. **Make sure the `LD_LIBRARY_PATH` environment variable points to the location of the Java virtual machine (JVM).**

   For example:

```
LD_LIBRARY_PATH=$JAVA_HOME/jre/lib/sparc/server;export LD_LIBRARY_PATH
```

2. **Start the region using the `-X` option to `unikixmain`.**

   The `-X` option requires that you specify the port number to be used by `unikixadmin`. It must be a unique port number. For example:

```
$ unikixmain -X 9876 other options
```

   When the `unikixadmin` server starts as part of region initialization, a message similar to the following is written to the `unikixmain.log` file:

```
08/30/2002 13:08:17 unikixmain :KIX0173I Process 1234 of type a started
```

The server advertises its startup with a message similar to the following:

```
08/30/2002 13:08:19 unikixadmin :KIX0301I Entering (VER. 8.0.0 - 08/28/2002
```

▼ To Register a Region With Sun MAA

1. **Start Sun MAT.**

   "Starting Sun MAT" on page 11.

2. **In the left tree pane, select the machine where the region you want to register is located.**

3. **On the menu, choose File → Register Region.**

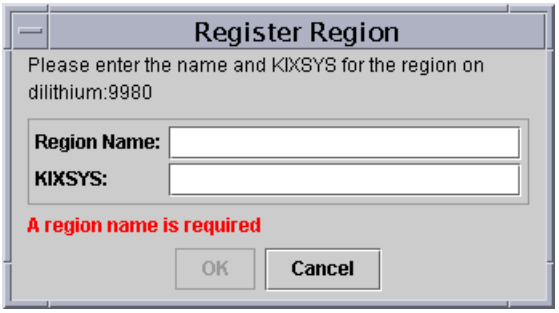

FIGURE 3-5 Registering a Region

4. **In the dialog box:**

   a. **Type the name of the region as you want it to display on the screen.**

   b. **Type the fully qualified directory name of the region's `$KIXSYS` directory.**

      The agent will determine the port number the `unikixadmin` daemon is listening on by accessing the `$KIXSYS` directory.

   c. **Click OK.**

An entry representing the region appears beneath the machine name in the left pane of the Sun MAT GUI. You can now use Sun MAT to perform administrative functions on the region. You might have to click the key icon next to the machine name to display the region.

Tip – If you register a region, but it does not seem to be available, verify that the region is running.

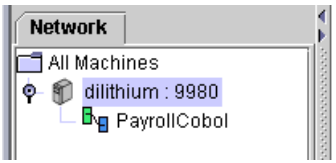

FIGURE 3-6 Region Entry

Note – Because the region is directly registered only with its agent, it immediately becomes visible to the user of any other instance of Sun MAT that has been configured to access the same agent.

The details about the region are stored by the agent to a local disk file (`etc/region_list.xml` under the Sun MAA installation directory), so that they will be available even after the agent is stopped and restarted.

FIGURE 3-7 shows a region that is accessible to Sun MAT.

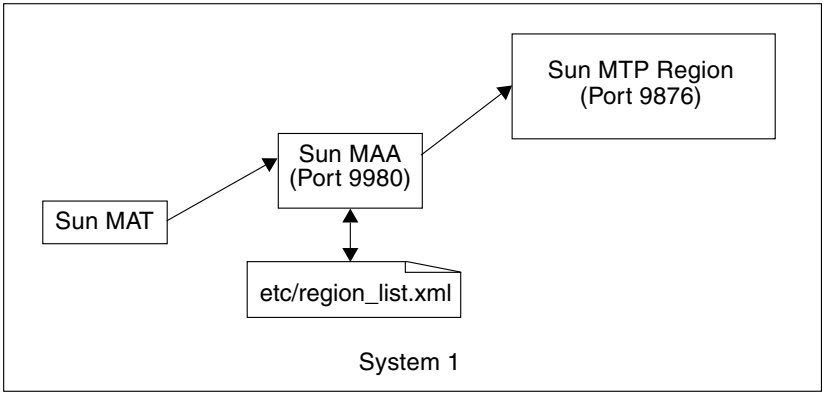

FIGURE 3-7 Access to a Region

Removing Regions and Machines

From time to time, your environment can change and you might want to deregister regions being monitored, or even remove a machine.

▼ To Deregister a Region

1. **In the list panel, select the region you want to deregister.**
2. **On the menu bar, choose File → Deregister Region.**

   Or right-click the region name in the list panel and select Deregister Region from the context menu.
3. **When the confirmation dialog box is displayed, click OK.**

   The region name is removed from the list pane and you can no longer monitor it.

▼ To Remove a Machine

1. **In the list panel, select the system you want to remove.**
2. **On the menu bar, choose File → Remove Machine.**

   Or right-click the system name and select Remove Machine from the context menu.
3. **When the confirmation dialog box is displayed, click OK.**

   The host is removed from the list pane and you are no longer able to monitor any regions running on it.

Setting Preferences

You can set the administration tool to control:

- The interval at which regions are polled for information. The actual updating of information may take longer, depending on your network.
- Start and exit behaviors.

▼ To Set Preferences

1. **On the menu bar, choose Settings → Preferences.**

   The Preferences dialog is displayed.

2. **Click the Polling tab to set a new polling interval.**

   The default is 15 seconds.

3. **Click the Start and Exit tab to change the default behaviors.**

   a. **In the Start area, the default behavior is to show the saved window information.**

      If you want to open the tool with a clear main area, check the Start with no windows option.

   b. **In the Exit area, the default behavior is to save the windows displayed in the main area.**

      Deselect the check box to not save the windows.

   The Save current layout button saves the layout currently displayed in the GUI and uses that layout the next time you open the tool, if you also selected Show saved window setup in the Start area of the dialog box.

CHAPTER 4

User Interface

The user interface contains the following elements:

- List Panel: Shows the regions available to monitor. It lists each machine and its registered regions.
- Main Area: Provides a multiple-document interface, allowing the user to look at multiple detail windows simultaneously.
- Detail Pane or Window: Shows specific information about a region or a resource of the region.
- Slider: Separates the list panel from the main area. The slider can be dragged to increase the space of the list panel or main area. The slider also has controls to minimize or maximize an area.
- Menu Bar: Contains global menu items. These items control the administration tool, not a detail pane, which has its own menu bar. See "Menu Bar" on page 23.
- Context Menus: Provide a quick way to access menu options by right-clicking an item, such as a machine name in the list panel, or a column heading in a window.

Menu Bar

The menu bar on the administration tool's main window contains items that control the administration tool.

File

The File menu is used to perform the following functions:

- Add Machine. See "Accessing Sun MTP Regions" on page 15.
- Register Region. See "Registering a Region With Sun MAA" on page 18.

- Deregister Region. Removes a region so that it cannot be monitored. See "To Deregister a Region" on page 21.
- Remove Machine. See "To Remove a Machine" on page 21.
- Exit. Exits the tool.

Window

The Window menu is used to perform these functions:

- Save current layout: Saves the layout of your windows so that they will appear in this way the next time you open the tool. See "Saving Window Layouts" on page 26.
- Cascade: Displays the detail windows in overlapping mode
- Tile: Tiles the detail windows side-by-side
- Restore All: Restores all detail windows to the previous view
- Minimize All: Minimizes all detail windows
- Maximize All: Maximizes all detail windows
- Close All: Closes all open detail windows

The bottom portion of the Window menu lists the open windows and displays a check mark next to the window that has focus.

▼ To Change Window Focus

- **At the bottom of the Window menu, click the check box next to the window you want to focus on.**

Chart

The Chart menu is used to create charts and gauges. See Chapter 6.

Settings

The Settings menu is used to set preferences for the administration tool. See "Setting Preferences" on page 22.

Help

The Help menu is used to display the About dialog, which provides version and copyright information about the administration tool.

Windows and How They Behave

When you open a window in the main area of the GUI—for example, by double-clicking a machine or region name—the window usually opens in a small format. The controls in the upper right corner of the title bar enable you to:

- Maximize the window
- Minimize the window
- Close the window

The GUI's multiple document interface enables you to display several windows at the same time by cascading or tiling them. Use the Window menu on the GUI's main menu bar to select the Cascade or Tile option.

Many of the windows also have a Show Details check box. When this box is checked, a detail pane appears below the scrolled list and displays all the datapoints for the selected item. This is useful if you have hidden some of the columns in a table.

Duplicating Windows

The administration tool allows you to create an exact copy of the detail window you are currently viewing. After the copy is made, the two windows are independent. Changes you make to one window do not affect the other. A duplicate window has exactly the same title as the source window.

Duplicating a window enables you to compare aspects of a region simultaneously. For example, you can open a window that shows all the programs defined for the region, and at the same time have another window showing the region's transactions.

▼ To Duplicate a Window

- **On the window's menu bar, select Window → Duplicate.**

  A copy of the window opens in the main area.

Saving Window Layouts

The administration tool enables you to save the window layouts you are currently viewing. When you save the layout, you save the state of all open windows. The state of a window includes its size and position, its current view, as well as selections you made on non-visible panes.

▼ To Save Window Layouts

- **On the main menu bar, select Window → Save current layout.**

  The next time you open the GUI, the window layout you saved is displayed.

  You can also save the current layout using the Settings → Preferences menu options, and clicking on the Start and Exit tab. See "Setting Preferences" on page 22.

Customizing Table Views

Most of the detail windows contain many columns of information—often too many to view at once in a meaningful way. You can customize the table so that you see only the information you want to see. You can select an individual column and hide it, or you can use the context menu to select the columns to display. You can also sort columns in ascending or descending order. The column heading that contains a triangle icon is the one controlling the sort order of the table.

▼ To Hide a Column

- **Right-click the column heading and select Hide Column.**

  To redisplay the hidden column, see "To Customize the Table View" on page 27.

▼ To Customize the Table View

1. **Right-click any column heading and select Choose Columns.**

   A dialog box listing each column heading, grouped into configuration or monitoring categories is displayed. Information in the configuration category is static; it changes only if you make a change to the region's configuration. Monitoring information is dynamic.

2. **Select the information you want to see and deselect any columns you want to hide.**

   You can use the Select All and Select None buttons to quickly select or deselect all columns.

   As you make your selections, the window shows your choices.

3. **When you are satisfied with your selections, click OK.**

▼ To Sort Column Information

1. **Right-click the heading of the column you want to sort.**
2. **Select either Sort Ascending or Sort Descending.**

CHAPTER 5

Machine and Region Windows

This chapter describes the machine and region windows.

Displaying the Machine Window

After you add a machine, you can display a window that contains summary information about the system and the regions running on it.

The machine window has two tabs:

- General: Displays information about the machine.
- MTP Regions: Displays a table containing information about each region running on the system. See "MTP Regions Tab" on page 31.

▼ To Display the Machine Window

- **Use either method:**
  - Double-click the machine name in the list panel.
  - Select the machine name, right-click, and choose Open from the context menu.

  The machine window opens and displays the General tab by default. The title bar contains the machine name and port number.

The datapoints on the General tab are:

TABLE 5-1 General Tab Datapoints

Datapoint	Description
Name	Name of the machine.
Port	Port number.
State	State of the machine. See "Machine States" on page 30.

Machine States

The icon and the color of the machine name indicate the state of the system:

- When a machine cannot be contacted, no regions appear below its name, and the machine name is red.
- When a machine can be contacted, all the regions registered with it are displayed in the list. You can view or hide the list of regions by clicking the key next to the machine icon.

Region States

A region can have one of the following states, which is denoted by an icon:

TABLE 5-2 Region States

Icon	State
	Can be contacted and running
	Can be contacted and not running
	Cannot be contacted When a region cannot be contacted, its name is red.

MTP Regions Tab

The machine window's MTP Regions tab lists each region on a host and a set of datapoints about that region.

▼ To Display the Regions Running on a Host

1. **Double-click the machine name in the list panel.**
2. **In the machine window, click the MTP Regions tab.**

   The table of regions is displayed.
3. **Select the region whose details you want to view.**
4. **Click the Show Details box in the lower left corner of the window.**

   A panel listing the table details in a more readable format is displayed. You can resize this panel by dragging its top slider.

   The datapoints on the region window are as follows:

TABLE 5-3 MTP Regions Tab Datapoints

Datapoint	Description
Name	Name of region, as defined when the region was registered.
Region Home	The path name of the region's `$KIXSYS` directory.
Running	Indicates if the region is running. Values are `true` and `false`.
System CPU	Total system processor time (CPU) for all transaction processors (in seconds). See "System CPU" on page 32.
Transaction Rate	Current transaction rate per second. See "Transaction Rate" on page 32.
Transactions Executed	Total number of transactions executed. See "Transactions Executed" on page 33.
Tx Classes	The total number of transaction classes configured for this region. The value includes the two reserved transaction classes KIXADMIN and KIXDFLT.

TABLE 5-3 MTP Regions Tab Datapoints *(Continued)*

Datapoint	Description
Tx Servers Configured	Number of transaction servers configured for this region
User CPU	Total user processor time (CPU) for all transaction processors (in seconds). See "User CPU" on page 33.
Users	Number of users connected to the region. See "Users" on page 34.

System CPU

Description

The System CPU utilization represents the amount of time spent executing system calls on behalf of a given region.

Values

The value is in the format: *hh:mm:ss*.

The time displayed is not very precise, because it is measured in units of seconds.

Interpretation

This value tells the administrator how much System CPU a region requires to process transactions. If this datapoint and the User CPU datapoint add up to actual wall clock time, the machine's hardware configuration may be saturated. In this case, either application optimization or additional hardware might be required.

Transaction Rate

Description

The Transaction Rate value shows the average number of transactions executed in a second by a region.

Values

The value is computed as follows:

#Transactions executed (since last update) divided by the *Update Interval*

Interpretation

This value gives you an idea of a particular region's throughput. This value can be more accurately computed by using a shorter polling interval. However, a shorter polling interval requires substantially more processing overhead.

The value of this datapoint can be influenced by many factors external to the region. For example, a development server's activities will cause a region to run slower than if the region has a dedicated machine.

Transactions Executed

Description

The Transactions Executed datapoint represents the cumulative total of transactions executed since the region was started.

Values

This value is retrieved directly from Sun MTP.

Interpretation

This value is used to compute the transaction rate for a particular region.

As long as your system is operating properly, and users are performing transactions, this datapoint should continue to increase. When this datapoint stops growing, check to make sure that all the transaction servers are idle and that no one is logged on to the system. If that is not the case, search for hang conditions.

User CPU

Description

The User CPU datapoint represents the amount of CPU time spent executing application code (not operating system calls) on behalf of a region.

Values

The value is in the format: *hh:mm:ss*.

The time displayed is not very precise, because it is measured in units of seconds.

Interpretation

This value tells the administrator how much User CPU a region requires to process transactions. If this datapoint and the System CPU datapoint add up to actual wall clock time, a machine's hardware configuration may be saturated. If this is the case, you must either optimize your application or consider adding additional hardware.

Users

Description

The Users datapoint shows the total number of users that are currently logged in to a region.

Values

This number ranges from 0 to the maximum number of users licensed for this region. It goes up and down depending on users logging in and out.

Interpretation

If this number is consistently close to the licensed maximum number of users, you should consider increasing your license since the region will disallow logins once the maximum licensed number is reached.

CHAPTER 6

Graphs and Gauges

The Sun MAT software can graphically display some of the data it collects from the regions it is monitoring. Data can be displayed as a graph or a gauge.

- A graph is a time-based display of one or more datapoints. See "Creating Graphs" on page 35.
- A gauge is a dynamic display of one or more datapoints that is not time-based. See "Creating Gauges" on page 39.

Creating Graphs

Graphs are time-based displays of one or more datapoints.

▼ To Create a Graph

1. **On the Sun MAT main window menu bar, choose Chart.**

   The drop-down menu contains two items, New Graph and New Gauge.

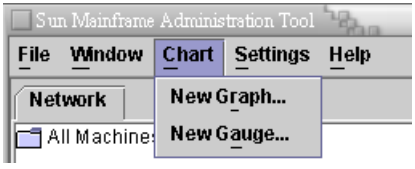

FIGURE 6-1 Chart Menu

2. **From the drop-down menu, choose New Graph.**

   The Configure the Graph dialog box is displayed. It has two tabs: General and Data.

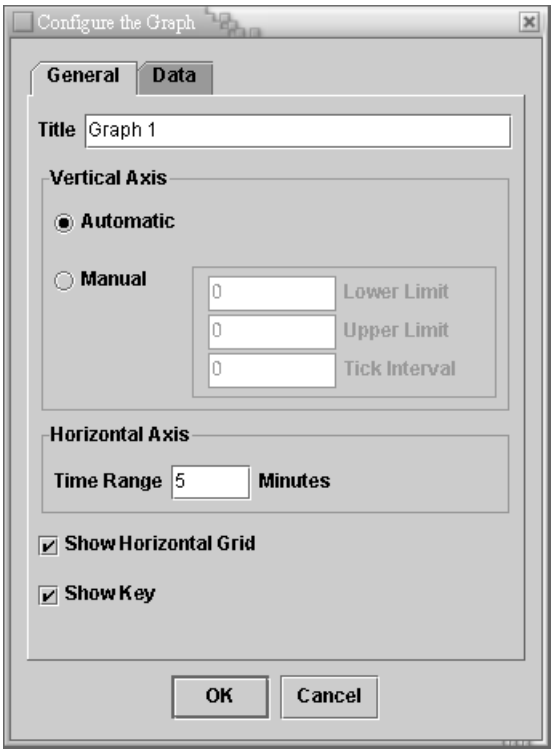

FIGURE 6-2 Graph Configuration—General Tab

Note – The Configure the Graph dialog can also be displayed by selecting Graph → Configure the Graph from the graph window. FIGURE 6-4 shows an example of a graph window.

---

3. **On the General tab, set the display values for the graph.**

   These values remain in force for all graphs until you change them.

   a. **In the Title field, type the name of the graph.**

   b. **In the Vertical Axis area, select either Automatic or Manual.**

      - If you select Automatic, the vertical axis is automatically scaled.
      - If you select Manual, you must specify the upper and lower limits of the range and the interval for each mark on the axis (tick). For example, if you are displaying the transaction rate, you might want to set the interval to 50 transactions.

   c. **In the Horizontal Axis area, type the time period in minutes that will be displayed on the graph.**

d. Select whether to display a horizontal grid on the graph.

e. Select whether to display the key for the graph.

4. Click the Data tab to display its contents.

The Data tab is used to specify the statistics that will be graphed from the available regions. When you configure a graph for the first time, the Charted Statistic area is empty. See FIGURE 6-3.

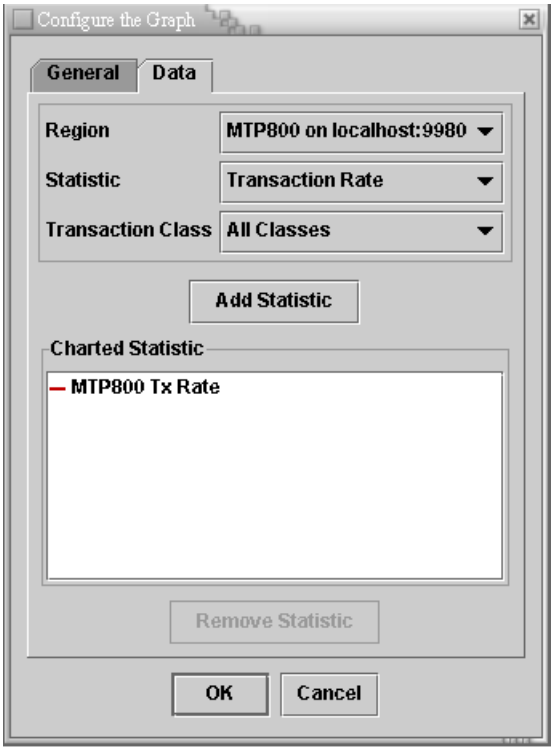

FIGURE 6-3 Graph Configuration—Data Tab

5. On the Data tab, choose the data you want to graph:

a. Click the drop-down arrow to select a region.

The region drop-down list shows all of the regions that are currently being monitored.

b. Click the drop-down arrow to select a statistic.

You can choose Transaction Rate, Active Tasks, Users, or Transactions Waiting from the list.

c. **Click the drop-down arrow to select a transaction class defined in the region or to select all transaction classes.**

6. **Click the Add Statistic button to add the data statistic you just defined to the Charted Statistic list.**

   To remove a statistic, select it, and click the Remove Statistic button.

7. **Repeat steps 5 and 6 to add other statistics to your graph.**

8. **Click OK to commit the changes you made to the configuration window and display the graph.**

   See FIGURE 6-4 for an example of a graph.

The Graph Window

The following figure shows a graph of transaction rates for the MTP800 region over a five-minute period. The graph displays the overall transaction rate and the transaction rates for the TCLASS1 and KIXDFLT transaction classes.

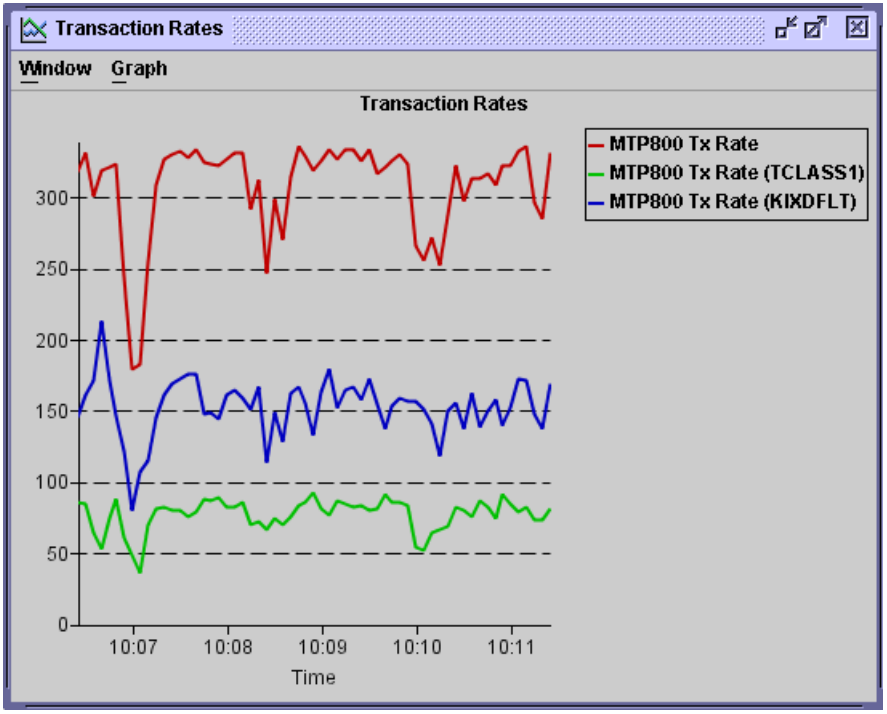

FIGURE 6-4 Graph Window—Example

The graph window menu has two items: Window and Graph, which have the following options:

- The Window menu contains options to duplicate the graph window and to close the graph window.
- The Graph menu contains options to configure the graph and to print the graph. If you select the Configure option, the Configure the Graph dialog box is displayed.

  As you make changes in the Configure the Graph dialog, the changes are dynamically displayed on the graph window. If you click OK, the changes are committed. If you press Cancel, all changes made in the current configuration session are backed out.

Creating Gauges

A gauge is a dynamic display of one or more datapoints that is not time-based.

▼ To Create a Gauge

1. **On the Sun MAT main window menu bar, choose Chart.**

   The drop-down menu contains two items, New Graph and New Gauge.

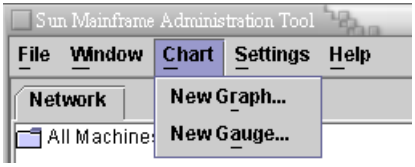

FIGURE 6-5 Chart Menu

2. **From the drop-down menu, choose New Gauge.**

   The Configure Gauge dialog box is displayed. It has two tabs: General and Data.

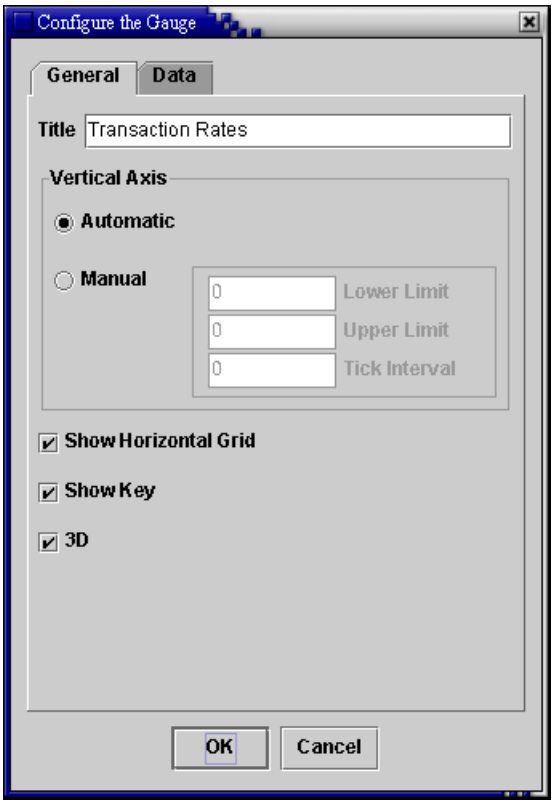

FIGURE 6-6 Gauge Configuration—General Tab

Note – The Configure the Gauge dialog can also be displayed by selecting Gauge → Configure the Gauge from the gauge window. FIGURE 6-8 shows an example of a gauge window.

---

3. **On the General tab, set the display values for the gauge.**

   These values remain in force for all gauges until you change them.

   a. **In the Title field, type the name of the gauge.**

   b. **In the Vertical Axis area, select either Automatic or Manual.**

      - If you select Automatic, the vertical axis is automatically scaled.
      - If you select Manual, you must specify the upper and lower limits of the range and the interval for each mark on the axis (tick). For example, if you are displaying the transaction rate, you might want to set the interval to 50 transactions.

c. **Select whether to display a horizontal grid on the gauge.**

d. **Select whether to display the key for the gauge.**

e. **Select whether to display the gauge three-dimensionally (3D check box).**

4. **Click the Data tab to display its contents.**

   The Data tab is used to specify the statistics that will be displayed from the available regions. When you configure a gauge for the first time, the Charted Statistic area is empty.

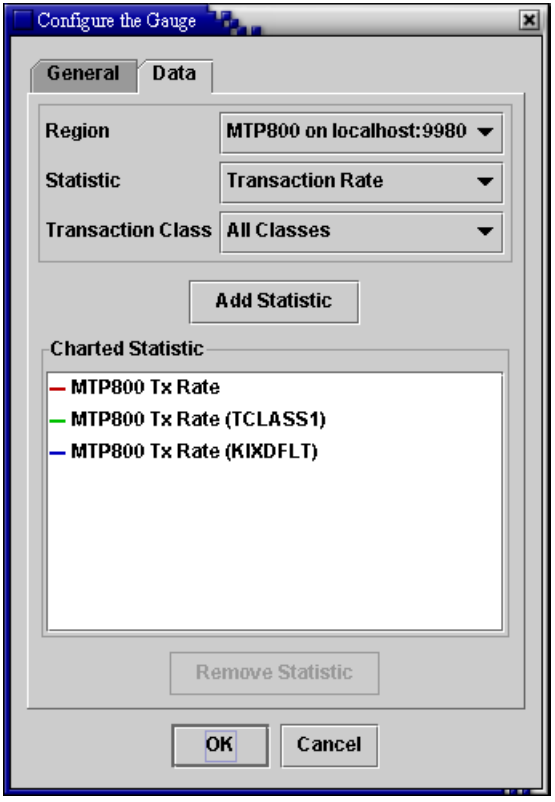

FIGURE 6-7 Gauge Configuration—Data Tab

5. **On the Data tab, choose the data you want to display:**

   a. **Click the drop-down arrow to select a region.**

      The region drop-down list shows all of the regions that are currently being monitored.

b. Click the drop-down arrow to select a statistic.

You can choose Transaction Rate, Active Tasks, Users, or Transactions Waiting from the list.

c. Click the drop-down arrow to select a transaction class defined in the region or to select all transaction classes.

6. **Click the Add Statistic button to add the data statistic you just defined to the Charted Statistic list.**

   To remove a statistic, select it, and click the Remove Statistic button.

7. **Repeat steps 5 and 6 to add other statistics to your gauge.**

8. **Click OK to commit the changes you made to the configuration dialog and display the gauge.**

   See FIGURE 6-8 for an example of a gauge.

The Gauge Window

The following figure shows a 3D gauge of transaction rates for the MTP800 region. The gauge displays the overall transaction rate and the transaction rates for the TCLASS1 and KIXDFLT transaction classes.

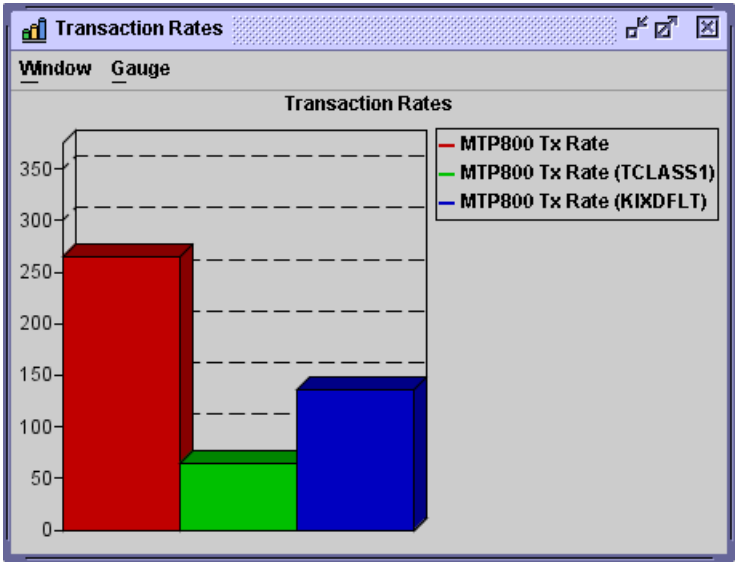

FIGURE 6-8 Gauge Window

The gauge window menu has two items: Window and Gauge, which have the following options:

- The Window menu has options to duplicate the gauge window and to close the gauge window.
- The Gauge menu has options to configure the gauge and to print the gauge. If you select the Configure option, the Configure the Gauge dialog box is displayed.

  As you make changes in the Configure the Gauge dialog, the changes are dynamically displayed on the gauge window. If you click OK, the changes are committed. If you press Cancel, all changes made in the current configuration session are backed out.

CHAPTER 7

Monitoring Regions Overview

This chapter describes how to display a region's window and the Overview tab. Refer the appropriate chapter for descriptions of the System, Communications, Resources, and Advanced tabs.

Displaying a Region's Window

To perform detailed monitoring of a region, display that region's window.

▼ To Display a Region's Window

1. **Select the region name in the list panel.**
2. **Use either method:**
   - Right-click and select Open from the context menu.
   - Double-click the region name in the list panel.

   See FIGURE 7-1.

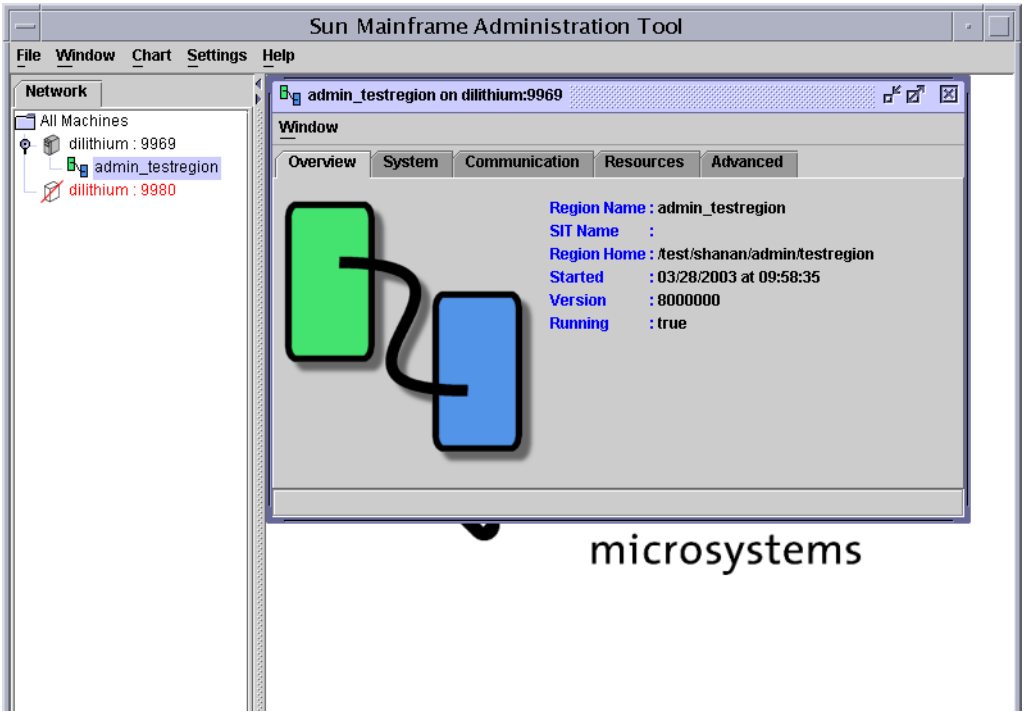

FIGURE 7-1 Region Overview Screen

The region window has five tabs:

- Overview tab: Contains general information about the region. See "Overview Tab" on page 47.
- System tab: Contains performance and other system-level information. See Chapter 8.
- Communications tab: Contains information about the region's communications servers. See Chapter 9.
- Resources tab: Contains information about the resources configured for the region, such as program, maps, and transactions. See Chapter 10.
- Advanced tab: Contains information primarily used by service personnel for diagnosing performance problems. See Chapter 11.

▼ To Display a Tab in a Region's Window

- **In the region window, click the tab you want to display.**

Overview Tab

The Overview Tab of the region window contains the following information:

TABLE 7-1 Overview Tab Datapoints

Datapoint	Description
Region Name	Identifies the region as it was registered to Sun MAT.
SIT Name	Identifies the region name as defined in the System Initiation Table (SIT).
Region Home	Identifies the path to the region's `$KIXSYS` directory as it was defined when the region was registered.
Started	Indicates the date and time the region was last started.
Version	Indicates the version of Sun MTP the region is using.
Running	Indicates if the region is running. Values are `true` or `false`.

CHAPTER 8

System Tab

The System tab displays performance information about the region, as well as configuration information.

▼ To Select the Type of Information to Display

1. **Open a region window and select the System tab.**

   See "Displaying a Region's Window" on page 45.

2. **Click the drop-down arrow at the top of the System tab.**

3. **Select the topic you want to view.**

   The following topics are available from the System tab:

   - "Performance Overview" on page 50
   - "Performance Details" on page 51
   - "Processes" on page 53
   - "Logs" on page 54
   - "Recovery" on page 54
   - "Batch" on page 59
   - "Languages" on page 60
   - "Limits" on page 61
   - "Accounting" on page 62

Performance Overview

The Performance Overview window displays information related to overall system overhead, and on a per transaction class basis.

At the top of the pane, there are two datapoints that apply to the region:

- Users: The current number of users connected to the region
- Shared Memory Size: The size of region shared memory segment, in kilobytes (K)

The rest of the pane contains datapoints related to the transaction classes defined for the region. You can view the information for individual transaction classes or for all transaction classes. Use the drop-down list to select a transaction class or all transaction classes.

The following datapoints are also displayed:

TABLE 8-1 Performance Overview Datapoints

Datapoint	Description
CPU (Kernel)	Total system processor (CPU) time for all transaction processors (in seconds).
CPU (User)	Total user processor (CPU) time for all transaction processors (in seconds).
Peak Transaction Rate	Peak transaction rate per second. See "Peak Transaction Rate" on page 52.
Peak Transactions Waiting	Maximum number of transactions forced to wait for a transaction processor.
Transaction Processor Restarts	Total number of transaction processor restarts.
Transaction Rate	Current transaction rate per second. See "Transaction Rate" on page 52.
Transactions Aborted	Total number of transactions aborted.
Transactions Deadlocked	Total number of transactions that encountered a deadlock.
Transactions Executed	Total number of transaction executed. See "Transactions Executed" on page 52.
Transactions Waiting	Current number of transactions waiting to be processed.

Performance Details

The Performance Details panel displays details related to active transactions.

Three entries at the top of the pane show:

- Total number of transactions executed. See "Transactions Executed" on page 52.
- Transaction rate (per second). See "Transaction Rate" on page 52.
- Peak transaction rate. See "Peak Transaction Rate" on page 52.

The following datapoints are displayed on this pane:

TABLE 8-2 Performance Details Datapoints

Datapoint	Description
Task ID	Task identification number.
Before Images	Total number of before images written.
Blocking Buffer	Buffer number the transaction is waiting for.
CICS Command	Last executed CICS command.
CPU (Kernel)	Total system CPU overhead.
CPU (User)	Total user CPU overhead.
Current Locks	Number of locks currently held.
Dataset	Dataset currently being processed.
Elapsed Time	Elapsed time.
Flags	Transaction status flags.
Logical Writes	Total number of logical writes.
Physical Reads	Total number of physical reads.
Physical Writes	Total number of physical writes.
Process ID	Transaction processor process ID.
Program Name	Name of the currently executing program.
Remote System ID	Remote system assigned to this transaction.
Status	Status of this transaction.
Terminal ID	Terminal identifier.
TX Class	Transaction class.
Transaction ID	Transaction name of this task.

TABLE 8-2 Performance Details Datapoints *(Continued)*

Datapoint	Description
User	Current user.
Waiting	`True` if the task is in the waiting state.
Waits	Total waits.

Transactions Executed

Description

The Transactions Executed datapoint represents the cumulative total of transactions executed since the region was started.

Interpretation

This value is used to compute the transaction rate for a particular region. As long as your system is operating properly and users are performing transactions, this datapoint should continue to increase.

Transaction Rate

Description

The Transaction Rate value shows the average number of transactions executed in a second by a region.

Values

The value of this datapoint is computed as follows:

#*Transactions executed (since last update)* divided by the *Polling Interval*

Interpretation

This value gives you an idea of a particular region's throughput. This value can be more accurately computed by using a shorter polling interval. However, a shorter polling interval requires substantially more processing overhead.

Peak Transaction Rate

Description

Peak Transaction Rate shows the highest number of transactions per second (shown with the Transaction Rate datapoint) calculated for a region since connecting to the region.

Interpretation

The datapoint gives you an idea of what your peak processing performance can be. Your system operates with the most throughput when the Transaction Rate datapoint remains consistently near the Peak Transaction Rate.

However, if the Transaction Rate is consistently close to Peak Transaction Rate, it is possible that the region is running at very close to the maximum throughput of the system and additional resources might be required.

Because this value is only calculated when connected to a region, the value may not represent the actual highest transaction rate achieved by the region. That point might have been reached when the region was not being monitored.

Processes

The Processes pane of the System tab displays information about the processes running in the region. Any of the following processes can be executing:

`unikixmain`	`unikixadmin`
`unikixstrt`	`unikixtran0–`*n*
`unikixrcv`	`unikixsched0–`*n*
`unikixept`	`unikixtrin`
`unikixprt`	`unikixqm`
`unikixCommMgr`	`unikixsock`
`unikixtcp`	

For each process, the following information is displayed:

TABLE 8-3 Processes Window Datapoints

Datapoint	Description
Name	Process name
CPU (Kernel)	System processor overhead in seconds
Process ID	Process identifier
CPU (User)	User processor overhead in seconds

Logs

The Logs pane enables you to select and view the following system logs:

- Main log, which is the `unikixmain.log` file
- Error log, which is the `unikixmain.err` file
- Debug log, which is the `unikixmain.dbg` file

▼ To Display a Log File

- **Click the radio button next to the log file you want to view.**

  The log file contents are displayed in the scrollable window.

Recovery

The Recovery window displays the configuration and status of the recovery file. This window displays the following statistics:

TABLE 8-4 Recovery Window Datapoints

Datapoint	Description
Configured	Indicates if recovery is configured. Values are `true` or `false`.
Environment	Environment variable that identifies the file's location. See "Environment" on page 56.
File System	Indicates if the region uses the standard UNIX file recovery method or the Native Recovery File System (NRFS). See "File System" on page 58.
Size	Size of the recovery file in blocks. See "Size" on page 57.
Last Wrap	Last time the (circular) recovery file wrapped. See "Last Wrap" on page 55.
Timeouts	Number of times Sun MTP flushes the file due to a timeout. See "Timeouts" on page 58.

TABLE 8-4 Recovery Window Datapoints *(Continued)*

Datapoint	Description
File Usage	Amount of the file used to store recovery information. See "File Usage" on page 56.
Reads	Total number of reads from the recovery file.
Writes	Total number of writes to the recovery file.

If recovery is active for a region, the Recovery window displays recovery attributes specified in the VSAM Configuration Table (VCT). If recovery is not turned on in the VCT, the Configured datapoint is false and the rest of the datapoints are blank. The minimum required size for a recovery file is 100 blocks. To activate recovery, you must set the size to at least 100 blocks in the VCT and restart the region.

One of the most important datapoints to monitor is the Last Wrap time. This datapoint displays the last time the recovery file became full, usually causing transactions to abort and the database file to roll back. If this condition persists, you should analyze the transactions that are causing this condition. They may not be properly accessing the database or syncpointing. In many cases, batch programs can cause the recovery file to wrap due to syncpointing the database at inappropriate intervals. Another reason for recovery file wrap may be that the recovery file is too small. Increase the size and determine if the Last Wrap value is similar to when the file was smaller.

Last Wrap

Description

The Last Wrap datapoint indicates the last time the recovery file became totally full.

Values

If a wrap has not occurred, a zero (0) is displayed. When a wrap occurs, this datapoint displays the date and time of the last wrap.

Interpretation

In Sun MTP, the recovery file is implemented as a circular file. When the file is full, it means records can no longer be written to the file because the tail has caught up with the head of the file. In this case, a transaction must be rolled back to free up the

recovery file for further processing. When this happens, the Last Wrap datapoint changes to display the exact time when the wrap occurred. After the rollback occurs, records are again written to the recovery file.

The Last Wrap datapoint must be monitored continuously (and compared to the current time) to catch multiple wraps. When such a condition persists, there are two possible causes:

- The recovery file might be too small. This is often the case when first deploying an application. You can increase the size of the file in the VCT and restart the region. Use the Last Wrap datapoint in conjunction with the % Used datapoint to quickly find the optimum recovery file size.
- Too much abnormal activity from the application programs. Analyze the programs that were aborted to determine if there are deficiencies in the logic synchronizing their database(s).

File Usage

Description

The File Usage datapoint displays the current amount of space (% of the total file space) required in the recovery file for processing before images. The value fluctuates to reflect the amount of output activity generated by a region.

Values

The File Usage datapoint is a number between 0 and 100, displayed to two decimal places. This datapoint is blank when recovery is not active.

Interpretation

The value of File Usage indicates how a particular recovery file's size is handling the traffic generated by a region. When the value stays close to 100%, and the Last Wrap datapoint continues to change, the recovery file size should be increased.

Many things can affect recovery file usage: application output to disk, deadlocks, looping transactions, program aborts, and so on. If the value continues to indicate the need to increase the size of the recovery file as you actually increase the file's size, examine the application programs for problems. A well-behaved set of applications should have an optimum recovery file size.

Environment

Description

The Environment datapoint displays the name of the environment variable that points to the directory where the recovery file is located. The administrator specifies this environment variable in the VCT.

Values

The value of the environment variable can be retrieved on the system running the region using the following command:

```
$ echo $variable-name
```

where *variable-name* is the value displayed by the Environment datapoint. Note that this datapoint is blank when recovery is not active.

Interpretation

The value of Environment must be a valid directory path name. The full path name of the recovery file is the value of the Filename datapoint concatenated to the end of the value of the Environment datapoint. The recovery file can be easily moved by changing the value of Environment in the VCT.

The Environment datapoint has no meaning for a native recovery file. For such a file, the Filename datapoint specifies the complete path name of the actual device (the device name is a file in the file system). Refer to the *Sun Mainframe Transaction Processing Software Configuration Guide* for information about configuring the Native Recovery File System (NRFS).

Size

Description

The Size datapoint indicates the size of the recovery file as configured for the region.

If the size is less than one hundred blocks, Sun MTP does not perform VSAM recovery.

Values

The Size datapoint displays the size of the recovery file in blocks, not bytes. Each block contains the same number of bytes as the VSAM block size for the system. The default VSAM block size is 4 KBytes. However, the block size can be 4, 8, 16, or 32 KBytes. If recovery is not active, the value is zero (0).

Interpretation

The optimum recovery file size depends on the recovery activity generated by applications running within a region. Logically, the recovery file is treated as a circular file. The actual file is a fixed size. In order to safely provide data integrity, Sun MTP must abort a transaction when the logical tail of the file catches up with the head. The value of the Last Wrap datapoint gives an idea of how often this happens. When the Last Wrap datapoint continues to change, recovery file size is a limiting factor in system performance, and the size should be increased.

One way to calculate a working recovery file size is to base the size on the activity of the transaction that performs the most writes. Each write to a database might generate an accompanying before image to the recovery file (if a successive write is to a different record). Multiply the number of these writes by the number of transaction servers. Then multiply that number by the average record size. Use the result as an initial recovery file size (remember to translate the size from bytes to blocks). Adjust the value when the Last Wrap datapoint shows too much activity.

Tip – You must increase the size of the recovery file more if you are using batch jobs that also require recovery file space.

Timeouts

Description

The Timeouts datapoint displays the number of times Sun MTP flushes before images when an internal (and currently non-configurable) timeout period expires and one of many trigger conditions have not already caused a flush. Sun MTP optimizes recovery file writing by grouping before images together and flushing them all at once. There are a few trigger conditions that cause a flush: reaching a maximum combined size (32 KBytes), reaching a command-line specified number of entries to flush, or experiencing transaction conditions like a syncpoint or end of transaction.

Values

The timeouts datapoint is an integer value initialized to zero at region startup.

Interpretation

The algorithm for writing out before images favors very active systems. You can expect the Timeouts value to be quite low for systems that perform a lot of disk activity. The number of timeouts should increase when a system does not perform very many transactions.

If the Timeouts count increases while the system is quite active, inform your authorized Sun service provider.

File System

Description

This datapoint identifies how I/O to a recovery file is managed. The type of management is determined when the recovery file is configured.

Values

Value	Description
`UNIX`	When the recovery file is a UNIX file, it is managed by the host file system just like any other file. The file is identified by concatenating the Filename value to the value of the Environment variable.
`Native`	Native recovery is an alternative method of recovery for a region. A native recovery file resides somewhere on a raw disk partition managed exclusively by Sun MTP. I/O efficiency might be improved with a native recovery file because Sun MTP bypasses overhead inherent in file system management.

Interpretation

On UNIX systems, a native recovery file might offer superior performance over a normal UNIX file. However, the initial setup is more involved for a native file. First, you must choose a free disk area. This could be a completely free raw partition (one not used by a file system), or an unused portion of a raw partition. The space for a native recovery file must be formatted with the Sun MTP `kixnrfs` utility. Refer to the *Sun Mainframe Transaction Processing Software Configuration Guide* for more details. Because it is so easy to set up and resize a UNIX recovery file, you can establish an optimum recovery file size with a UNIX file first, then transfer the file to a native organization.

Batch

The Batch panel displays the following information:

TABLE 8-5 Batch Window Datapoints

Datapoint	Description
Batch System	Node name of the configured batch system. See "Batch System" on page 59.
Batch Search Interval	Frequency with which the region checks to determine if a batch job is waiting to run

Batch System

Description

A region can be configured to connect to a Sun MBM node. This connection allows Sun MBM batch jobs to access the region's VSAM datasets. The Batch System datapoint identifies the node directory, which is configured in the region's VCT.

Values

If a node directory is defined, that value is displayed in this datapoint. Otherwise, the value is `N/A`.

Interpretation

If the region is not configured, Sun MBM batch jobs cannot access VSAM datasets maintained in that region. A region configured for connection to a Sun MBM node does not guarantee a connection. The connection can be verified by examining system messages written to the region's log file, `$KIXSYS/unikixmain.log`.

Languages

The Languages pane displays the languages used in the programs running in the region.

A box with the language name is displayed in the pane as well as the following datapoints:

TABLE 8-6 Languages Window Datapoints

Datapoint	Description
Configured	Indicates if a language is configured for the region. Values are `true` and `false`.
Location	Identifies the directory path where the language is installed.
Version	Indicates the version of the language.

Limits

The Limits panel displays a set of configuration restrictions:

TABLE 8-7 Limits Window Datapoints

Datapoint	Description
Maximum Background Tasks	Indicates the maximum number of background tasks configured.
Maximum Batch Jobs	Indicates the number of batch jobs that can run concurrently in the region.
Maximum Debug Terminals	Indicates the number of debug terminals configured. See "Maximum Debug Terminals" on page 61.
Transaction Processors	Indicates the number of transaction servers configured. See "Transaction Processors" on page 61.
Maximum Users	Indicates the number of concurrent users configured.
Maximum Query Jobs	Indicates the number of query jobs that can run concurrently in the region.
Terminal Timeout	Indicates the terminal timeout in seconds.

Transaction Processors

Description

The Transaction Processors datapoint displays the number of transaction servers available for executing transactions. Each transaction server is a system process.

Values

The number of Transaction Processors is the number of transaction servers configured for the region.

Maximum Debug Terminals

Description

The Maximum Debug Terminals datapoint indicates how many debug terminals are available for debugging a region.

Values

The value of Maximum Debug Terminals is the number configured for the region.

Interpretation

This datapoint represents the number of transaction servers to be used for executing transactions in debug mode.

If this value is zero, no debugging can be performed.

Accounting

The Accounting panel displays a set of configuration options.

TABLE 8-8 Accounting Window Datapoints

Datapoint	Description
System Accounting	Indicates whether system-level accounting is enabled. Values are `true` or `false`.
Transaction Accounting	Indicates whether default transaction accounting is enabled. Values are `true` or `false`.
User Accounting	Indicates whether default user accounting is enabled. Values are `true` or `false`.

CHAPTER 9

Communications Tab

The Communications Tab displays information about each of the following communications servers:

TCP/IP ISC Server

The TCP/IP ISC Server window displays information about the Sun MTP intersystem communications (ISC) over TCP/IP system daemon (`unikixtcp`). Refer to the *Sun Mainframe Transaction Processing Software Administrator's Guide* and the *Sun Mainframe Transaction Processing Software Configuration Guide* for information about configuring and using TCP/IP communications in a region.

The datapoints on this window are listed in the following table:

TABLE 9-1 TCP/IP Server Datapoints

Datapoint	Description
3270 Messages Received	Indicates the total number of 3270 messages received.
Client Requests	Indicates the total number of client requests.
Closed Sockets	Indicates the total number of closed sockets.

TABLE 9-1 TCP/IP Server Datapoints *(Continued)*

Datapoint	Description
Complete Sends	Indicates the total number of complete sends to clients.
Configured	Indicates if the TCP/IP server is configured. Values are `true` or `false`.
Connects From MTP	Indicates the total number of connects from Sun MTP regions.
Connects From MTP Client	Indicates the total number of connects from the Sun MTP Client.
CPU (Kernel)	Indicates the total kernel processor (CPU) time used by this server. This is the amount of time spent executing code within the operating system kernel.
CPU (User)	Indicates the total number of seconds of user time used by this server. This is the amount of time spent executing code within the server as opposed to the operating system kernel.
Event Requests	Indicates the total number of event requests from Sun MTP.
Event Requests (`CONVERSE`)	Indicates the total number of `CONVERSE` event requests from Sun MTP.
Event Requests (`ENDTRAN`)	Indicates the total number of `ENDTRAN` event requests from Sun MTP.
Event Requests (`SEND`)	Indicates the total number of `SEND` event requests from Sun MTP.
Inbound Sessions	Indicates the maximum number of concurrent ISC requests that can be accepted by the region using the TCP/IP protocol.
Incomplete Receives	Indicates the total number of incomplete receives from clients.
Incomplete Sends	Indicates the total number of incomplete sends to clients.
Maximum Concurrent Sockets	Indicates the maximum number of sockets opened at any one time.
Maximum Queued Requests	Indicates the maximum number of queued requests at any one time.
Orphaned Sessions	Indicates the total number of orphaned sessions.
Outbound Sessions	Indicates the total number of outbound sessions. See "Outbound Sessions" on page 65.
Port	Identifies the port number that the region uses to process ISC requests over the TCP/IP protocol.
Process ID	Identifies the process ID assigned to this process by the operating system.
Queued Requests	Indicates the total number of queued requests.

TABLE 9-1 TCP/IP Server Datapoints *(Continued)*

Datapoint	Description
Queued Requests Executed	Indicates the total number of queued requests that have been executed.
Queued Requests Transferred	Indicates the total number of queued requests that have been transferred.
Requests	Indicates the total number of requests from Sun MTP.
Signals	Indicates the total number of signals handled.
Transactions Replied	Indicates the total number of transaction reply messages sent to Sun MTP.
Transactions Submitted	Indicates the total number of transactions submitted to Sun MTP.
Waits	Indicates the number of times an application waited for an outbound session. See "Waits" on page 65

Outbound Sessions

Description

The maximum number of concurrent ISC requests that can be initiated by Sun MTP applications.

Interpretation

If this value is too small, application programs might be delayed. See "Waits" on page 65.

Waits

Description

The number of times an application program was forced to wait for an available outbound session.

Interpretation

A nonzero value indicates that the number of outbound sessions is too small. Increase the value of the outbound sessions, using the `TCPSTERM` environment variable, to support the number of concurrent ISC applications that can execute in the region.

SNA ISC Server

The SNA ISC Server pane displays information about the Sun MTP intersystem communications (ISC) over SNA system daemon (`unikixdcl`). Refer to the *Sun Mainframe Transaction Processing Software Administrator's Guide* and the *Sun Mainframe Transaction Processing Software Configuration Guide* for information about configuring and using SNA communications in a region.

The datapoints on this window are listed in the following table:

TABLE 9-2 SNA Server Datapoints

Datapoint	Description
Configured	Indicates if the SNA server is configured. Values are true or false.
Debug	Indicates if debugging is enabled. Values are true or false.
CPU (Kernel)	Identifies the total kernel processor (CPU) time used by this server. This is the amount of time spent executing code within the operating system kernel.
LU Name	Identifies the local LU name of this server
Process ID	Identifies the process ID assigned to this process by the operating system.
Inbound Sessions	Indicates the maximum number of concurrent ISC requests that can be accepted by the region using the SNA protocol.
Outbound Sessions	Indicates the total number of outbound sessions. See "Outbound Sessions" on page 66.
Waits	Indicates the number of times an application waited for an outbound session. See "Waits" on page 67.
CPU (User)	Indicates the total number of seconds of user time used by this server. This is the amount of time spent executing code within the server as opposed to the operating system kernel.

Outbound Sessions

Description

The maximum number of concurrent ISC requests that can be initiated by Sun MTP applications.

Interpretation

If this value is too small, application programs might be delayed. See "Waits" on page 67.

Waits

Description

The number of times an application program was forced to wait for an available outbound session.

Interpretation

A nonzero value indicates that the number of outbound sessions is too small. Increase the value of the outbound sessions, using the `DCLSTERM` environment variable, to support the number of concurrent ISC applications that can execute in the region.

PU4/5 Server

The PU4/5 server enables users of 3270 SNA devices to connect to a Sun MTP region. Refer to the *Sun Mainframe Transaction Processing Software Configuration Guide* for information about configuring the PU4/5 server.

The datapoints on this window are:

TABLE 9-3 PU4/5 Server Datapoints

Datapoint	Description
Configured	Indicates if the PU4/5 server is configured for the region. Values are `true` or `false`.
CPU (Kernel)	Indicates the total number of seconds of system time used by this server. This is the amount of time spent executing code within the operating system kernel.
CPU (User)	Indicates the total number of seconds of user time used by this server. This is the amount of time spent executing code within the server as opposed to the operating system kernel.
Process ID	Identifies the process identifier assigned by the operating system.

TN3270 Server

The TN3270 server (`unikixtnemux`) process supports 3270 emulators using the TCP/IP TN3270 and TN3270E protocols. Refer to the *Sun Mainframe Transaction Processing Software Configuration Guide* for information about configuring the TN3270 server.

The following datapoints are displayed on the TN3270 Server window:

TABLE 9-4 TN3270 Server Datapoints

Datapoint	Description
Configured	Indicates if the server is configured for the region. Values are `true` and `false`.
CPU (Kernel)	Indicates the total number of seconds of system time used by this server. This is the amount of time spent executing code within the operating system kernel.
CPU (User)	Indicates the total number of seconds of user time used by this server. This is the amount of time spent executing code within the server as opposed to the operating system kernel.
Debug	`True` when the server is in debug mode.
Keyboard Reset	Indicates whether the TN3270 server will ensure that a keyboard reset command is issued.
Login	`True` when a user login is required.
Maximum Endpoints	Indicates the maximum number of endpoints each TN3270 server can manage.
Port	Identifies the starting port number that the region uses to process TN3270 requests.
Process ID	The process identifier assigned by the operating system.
Processes	Indicates the number of TN3270 processes configured at the region.

MQ Server

The MQ server process (`unikixqm`) acts as a queue manager server for WebSphere MQ allowing applications to ship messages via MQ queues. Refer to the *Sun Mainframe Transaction Processing Software Configuration Guide* for information about configuring the MQ server process and the *Sun Mainframe Transaction Processing Software Developer's Guide* for information about using MQ with Sun MTP.

The following datapoints are displayed on the MQ server window:

TABLE 9-5 MQ Server Datapoints

Datapoint	Description
Configured	Indicates if the MQ server process is configured for the region. Values are `true` and `false`.
CPU (Kernel)	Indicates the total number of seconds of system time used by this server. This is the amount of time spent executing code within the operating system kernel.
CPU (User)	Indicates the total number of seconds of user time used by this server. This is the amount of time spent executing code within the server as opposed to the operating system kernel.
Process ID	Identifies the process identifier assigned by the operating system.
Queue Manager Location	Identifies the location of the MQ queue manager. See "Queue Manager Location" on page 70.
Queue Manager Name	Identifies the MQ queue manager. See "Queue Manager Name" on page 70.
Queue Name	Identifies the MQ trigger queue used to initiate a Sun MTP transaction. See "Queue Name" on page 71.
Requests	Indicates the number of MQ trigger messages processed since the region was started. See "Requests" on page 70.
Status	Indicates whether the MQ communications server is active and available to process requests. See "Status" on page 71.

Queue Manager Location

Description

The location of the WebSphere MQ queue manager.

Values

The value is obtained from the WebSphere MQ environment variable `MQSERVER`. If this variable is not defined, the value `N/A` is displayed.

Queue Manager Name

Description

This datapoint is the name of the MQ queue manager used by the `unikixqm` daemon to initiate a Sun MTP transaction when MQ messaging and queuing services are available.

Values

Queue Manager Name consists of 1–40 characters. If the name is blank, the queue manager being serviced is the default queue manager as defined to MQ.

Note – This datapoint is set to `N/A` when the MQ services are not available and the Configured datapoint displays a value of `false`.

Interpretation

This datapoint, along with the Queue Name value are sufficient to identify the MQ queue that is being monitored by the region when MQ services are available.

Requests

Description

This datapoint indicates the number of MQ trigger messages that have been processed by `unikixqm` since the region was started.

Values

The datapoint displays a positive integer value greater than or equal to zero.

This datapoint is a counter that is incremented every time a Sun MTP transaction is scheduled as a result of an MQ trigger message.

Status

Description

Indicates whether the MQ communications server is active and available to process requests.

Values

Value	Description
`0`	Indicates that the MQ server is not ready to process requests.
`1`	Indicates that the MQ server is ready to process requests.

Interpretation

A value of 0 might indicate that the MQ server is having difficulty connecting to the associated MQ queue manager. Check the Sun MTP error log for messages related to the MQ server.

Queue Name

Description

This datapoint is the name of the MQ trigger queue used by the `unikixqm` daemon to initiate a Sun MTP transaction when MQ messaging and queuing services are available.

Values

The Queue Name datapoint consists of 1–40 characters.

Note – This datapoint is set to `N/A` when the MQ services are not available and the Configured datapoint displays a value of `false`.

Interpretation

This datapoint, along with the Queue Manager Name value are sufficient to identify the MQ queue that is being monitored by the region when MQ services are available.

Socket Server

The Socket window displays statistics about requests made to the Sun MTP socket server (`unikixsock`). The Sun MTP socket facility allows application programs to make requests over a socket connection. The request must adhere to the IBM format. The socket server accepts three different types of requests:

- Immediate (IM), which establishes an immediate link to a transaction server
- Interval Control (IC), which places a transaction on the Sun MTP start queue
- Transient Data (TD), which writes data to a specific transient data queue, usually an intrapartition queue

The Socket window displays the following statistics:

TABLE 9-6 Socket Server Datapoints

Datapoint	Description
Configured	Indicates if the socket server is configured for the region. Values are `true` and `false`.
CPU (Kernel)	Indicates the total number of seconds of system time used by this server. This is the amount of time spent executing code within the operating system kernel.
CPU (User)	Indicates the total number of seconds of user time used by this server. This is the amount of time spent executing code within the server as opposed to the operating system kernel.
Denials	Indicates the number of socket requests that have failed the user exit function. See "Denials" on page 78.
IC Total Bytes	Indicates the total number of characters sent in the DATA field by all Interval Control (IC) requests to the socket server. See "IC Total Bytes" on page 73.
IC Total Failures	Indicates the number of unsuccessful Interval Control (IC) socket requests to a region's socket server. See "IC Total Failures" on page 74.
IC Total Requests	Indicates the number of successful Interval Control (IC) socket requests to a region's socket server. See "IC Total Requests" on page 74.
IM Total Bytes	Indicates the total number of characters sent in the DATA field by all Immediate (IM) requests to the socket server. See "IM Total Bytes" on page 75.
IM Total Failures	Indicates the number of unsuccessful Immediate (IM) socket requests to a region's socket server. See "IM Total Failures" on page 75.

TABLE 9-6 Socket Server Datapoints *(Continued)*

Datapoint	Description
IM Total Requests	Indicates the number of successful Immediate (IM) socket requests to a region's socket server. See "IM Total Requests" on page 76.
Port	Identifies the port number that the region uses to process socket requests.
Process ID	Contains the process identifier assigned by the operating system.
TD Total Bytes	Indicates the total number of characters processed by the socket server for Transient Data (TD) requests. See "TD Total Bytes" on page 76.
TD Total Failures	Indicates the number of unsuccessful Transient Data (TD) socket requests to a region's socket server. See "TD Total Failures" on page 77.
TD Total Requests	Indicates the number of successful Transient Data (TD) socket requests to a region's Socket server. See "TD Total Requests" on page 77.

The socket facility also enables you to customize an exit routine for a site. This routine can perform a security check to ensure that the socket facility only honors validated requests. The security exit routine either allows or refuses a request.

Refer to the *Sun Mainframe Transaction Processing Software Developer's Guide* for information about using sockets.

IC Total Bytes

Description

The IC Total Bytes datapoint shows the total number of characters sent in the `DATA` field by all Interval Control (IC) requests to the socket server. Refer to "IC Total Requests" on page 74 for a complete description of the IC request and its format.

Values

The IC Total Bytes datapoint is a cardinal integer count.

Interpretation

The IC Total Bytes datapoint is a cumulative total of the number of characters passed to the socket server in the `DATA` field of all IC socket requests. Only successful requests influence the IC Total Bytes count.

This datapoint provides a good idea of the amount of IC socket traffic being processed by a region. As long as data accompanies IC requests, the IC Total Bytes value should increase with the number of IC Total Requests.

IC Total Failures

Description

The IC Total Failures datapoint shows the number of unsuccessful Interval Control (IC) socket requests to a region's socket server.

Values

The IC Total Failures datapoint is a cardinal integer count.

Interpretation

An IC socket request from an application can fail for different reasons, for example:

- The `TRANSID` might not be present in the transaction table
- The `DATA` argument might exceed the 35-character limit

The region prints a message to `$KIXSYS/unikixmain.err` for each failure. The message is similar to the following:

```
KIX2346E Socket request of type IC failed with EIBRESP=28 TRANSIDERR
```

This datapoint provides a general idea of the volume of unsuccessful IC requests made to socket server. It does not, however, indicate why requests have failed. Examine the error log file to identify the cause of any failures. Note that Sun MTP cannot possibly record the number of failures that occur without a successful socket connection to the socket server. The application must first connect to the server for Sun MTP to recognize the existence of the request.

IC Total Requests

Description

The IC Total Requests datapoint shows the number of successful Interval Control (IC) socket requests to a region's socket server. An application makes an IC request when a delay is required before processing a given transaction.

Values

The IC Total Requests datapoint is a cardinal integer count.

Interpretation

The IC Total Requests datapoint is a cumulative total of the number of successful IC socket requests made to the region's socket server. It does not include the number of IC Total Failures.

This datapoint shows how many successful IC socket requests have been made to the socket server. As long as data accompanies IC requests, the IC Total Bytes datapoint should increase with the IC Total Requests.

IM Total Bytes

Description

The IM Total Bytes datapoint shows the total number of characters sent in the DATA field by all Immediate (IM) requests to the socket server. Refer to "IM Total Requests" on page 76 for a complete description of the IM request and its format.

Values

The IM Total Bytes datapoint is a cardinal integer count.

Interpretation

The IM Total Bytes datapoint is a cumulative total of the number of characters passed to the socket server in the DATA field of all IM socket requests. Only successful requests influence the IM Total Bytes count. Therefore, the IM Total Failures do not influence this datapoint.

This datapoint provides a good idea of the amount of IM socket traffic being processed by a region. As long as data accompanies IM requests, the number of IM Total Bytes should increase with the number of IM Total Requests.

IM Total Failures

Description

The IM Total Failures datapoint shows the number of unsuccessful Immediate (IM) Socket requests to a region's socket server.

Values

The IM Total Failures datapoint is a cardinal integer count.

Interpretation

An IM socket request from an application can fail for several reasons, for example:

- The TRANSID might not be present in the transaction table
- The DATA argument might exceed the 35-character limit

The region prints a message to $KIXSYS/unikixmain.err for each failure. The message is similar to the following:

```
KIX2346E Socket request of type IM failed with EIBRESP=28 TRANSIDERR
```

This datapoint provides a general idea of the volume of unsuccessful IM requests made to socket server. It does not, however, indicate why requests have failed. Examine the error log file to identify the cause of any failures. Note that Sun MTP cannot possibly record the number of failures that occur without a successful socket connection to the socket server. The application must first connect to the server for Sun MTP to recognize the existence of the request.

IM Total Requests

Description

The IM Total Requests datapoint shows the number of successful Immediate (IM) socket requests to a region's socket server. An application makes an IM request to execute a transaction with no delay. The transaction server and the application use the socket to communicate directly. The application, however, must make the request directly to the socket server.

Values

The IM Total Requests datapoint is a cardinal integer count.

Interpretation

The IM Total Requests datapoint is a cumulative total of the number of successful IM socket requests made to the socket server. It does not include the number of IM Total Failures.

This datapoint tells exactly how many successful IM socket requests have been made to the socket server. As long as data accompanies IM requests, the number of IM Total Bytes should increase with the number of IM Total Requests.

TD Total Bytes

Description

The Total TD Bytes datapoint shows the total number of characters processed by the socket server for Transient Data (TD) requests. Refer to "TD Total Requests" on page 77 for a complete description of the TD request and its format.

Values

The Total TD Bytes datapoint is a cardinal integer count.

Interpretation

The Total TD Bytes datapoint is a cumulative total of the number of characters passed to the socket server in the `DATA` field of all TD socket requests. Only successful requests influence the Total TD Bytes count. Therefore, failures do not influence this datapoint.

This datapoint provides a good idea of the amount of TD socket traffic is being processed by a region. As long as data accompanies TD requests, Total TD Bytes should increase with the number of Total TD Requests.

TD Total Failures

Description

The TD Total Failures datapoint shows the number of unsuccessful Transient Data (TD) socket requests to a region's socket server.

Values

The TD Total Failures datapoint is a cardinal integer count.

Interpretation

A TD socket request from an application can fail for different reasons, for example:

- The TD queue might not be defined in the Destination Control Table (DCT)
- The `DATA` field might exceed the 35-character limit

The region prints a message to `$KIXSYS/unikixmain.err` for each failure. The message is similar to the following:

```
KIX2346E Socket request of type TD failed with EIBRESP=28 TRANSIDERR
```

This datapoint provides a general idea of the volume of unsuccessful TD requests made to the socket server. It does not, however, indicate why requests have failed. Examine the error log file to identify the cause of any failures. Note that Sun MTP cannot possibly record the number of failures that occur without a successful socket connection to the socket server. The application must first connect to the server for Sun MTP to recognize the existence of the request.

TD Total Requests

Description

The TD Total Requests datapoint shows the number of successful Transient Data (TD) socket requests to a region's socket server. An application makes a TD request to write data to a particular TD queue, usually to work toward the firing of a trigger condition.

Values

The TD Total Requests datapoint is a cardinal integer count.

Interpretation

The TD Total Requests datapoint is a cumulative total of the number of successful TD socket requests made to the socket server. It does not include the number of failures.

This datapoint tells exactly how many successful TD socket requests have been made to the socket server. As long as data accompanies TD requests, the number of TD Total Bytes should increase with the number of TD Total Requests.

Denials

Description

The Denials datapoint displays the number of socket requests that have failed the user exit function. By default, the user exit routine allows all socket requests to continue. A site can customize this routine to put security restrictions on the types of requests available to the user community. Sun MTP provides the source code to the `kxsktxit()` routine. Site changes to it must be bound to the transaction server executable. Refer to the *Sun Mainframe Transaction Processing Software Administrator's Guide* for details.

Values

When shown, the Denials datapoint is a cardinal integer count.

Interpretation

The Denials datapoint is the total number of denied requests for all types of socket requests. Sun MTP counts the number of denials by keeping track of the return value from the `kxsktxit()` routine.

This datapoint shows the number of socket requests made by the user community that are not allowed by the administrator. Do not confuse the various failures datapoints with the Denials datapoint. The former counts the number of requests that fail because of incorrect request parameters. Denials are controlled by the administrative security exit routine, `kxsktxit()`.

Secure Socket Layer (SSL) Server

The SSL socket server (`unikixssl`) enables clients to communicate with a remote Sun MTP region using SSL. Refer to the *Sun Mainframe Transaction Processing Software Configuration Guide* for information about configuring the SSL server and the *Sun Mainframe Transaction Processing Software Developer's Guide* for information about using the SSL server.

The following datapoints are displayed on the SSL Socket Server pane:

TABLE 9-7 SSL Socket Server Datapoints

Datapoint	Description
Certificate	Identifies the nickname of the SSL certificate.
Configured	Indicates if the SSL server is configured for the region. Values are `true` and `false`.
Debug	Indicates if debug tracing is enabled. Values are `true` and `false`.
CPU (Kernel)	Indicates the total number of seconds of system time used by this server. This is the amount of time spent executing code within the operating system kernel.
Port	Identifies the port number that the region uses to process SSL requests.
Process ID	The process identifier assigned by the operating system.
CPU (User)	Indicates the total number of seconds of user time used by this server. This is the amount of time spent executing code within the server as opposed to the operating system kernel.

Administration Server

The Administration server (`unikixadmin`) is a daemon process that enables a region to be monitored by the administration tool.

The following datapoints are displayed on the Administration Server pane:

TABLE 9-8 Administration Server Datapoints

Datapoint	Description
Configured	Indicates if the Administration server is configured for the region. Values are `true` and `false`.
CPU (Kernel)	Indicates the total number of seconds of system time used by this server. This is the amount of time spent executing code within the operating system kernel.
Port	Identifies the port number that the region uses to process administration requests. The agent uses this port to communicate with the region. This port number is specified in the `-X` option to `unikixmain` when the region is started. See "To Enable the `unikixadmin` Server" on page 18.
PID	The process identifier assigned by the operating system.
Cache Refreshes	Indicates the total number of times the `unikixadmin` system daemon rebuilt its local cache. See "Cache Refreshes" on page 81.
Configuration Retrievals	Indicates the total number of requests for datapoint information classed as `config` type (consistent).
Monitoring Retrievals	Indicates the total number of requests for datapoint information classed as `monitor` type (volatile).
CPU (User)	Indicates the total number of seconds of user time used by this server. This is the amount of time spent executing code within the server as opposed to the operating system kernel.

Cache Refreshes

Description

The Cache Refreshes datapoint indicates the total number of times the `unikixadmin` system daemon rebuilt its local cache. The `unikixadmin` cache has a lifetime of one second. Multiple requests for datapoint information can be serviced from the cache if they arrive within the cache lifetime.

Interpretation

If the Cache Refreshes datapoint is less than the Configuration and Monitoring datapoints, it indicates that the cache is being used effectively.

CHAPTER 10

Resources Tab

The Resources tab enables you to display a window for each resource type you want to monitor. The following resource types are available:

CICS Commands

The CICS Commands pane displays a list of supported CICS commands. The entry for each command shows the number of times the command was issued by the application program(s) executing in the region. All counts are initialized to zero when the region is started.

The following datapoints are displayed on the CICS Commands pane:

TABLE 10-1 CICS Commands Window Datapoints

Datapoint	Description
Name	Identifies the CICS command. See "Name" on page 84.
Count	Indicates the number of times a command has been requested by application programs. See "Count" on page 84.

Name

Description

The Name datapoint shows a CICS command supported by Sun MTP.

Interpretation

To find out if a particular CICS command is executed in a region, look at the Count datapoint. If the Count datapoint shows a nonzero number, then the command has executed *n* times since system initiation.

Count

Description

The value of Count for a CICS command is the number of times a command has been requested by application programs.

Values

Count is an integer initialized to zero at region startup and incremented for each call to a particular command by an application program.

Interpretation

The column of Count values show cumulative counts from all application programs executing in a region.

The Count column gives a profile of the types and counts of CICS commands issued by the programs that run in a region.

Files

The Files window shows the datasets and alternate indexes available to the region's transactions and programs.

▼ To Display the Files Window

- **Click the down arrow on the Resources tab and select Files.**

  The table shows all the VSAM files configured for the region.

▼ To Display Information About a Particular Dataset or Alternate Index

- **Click the dataset or alternate index name.**

  The detail panel is displayed under the table.

  The following datapoints are displayed for each dataset:

TABLE 10-2 Files Window Datapoints

Datapoint	Description
Name	Identifies the dataset. See "Name" on page 86
Alternate Index	Indicates if an alternate exists for the dataset. See "Alternate Index" on page 87.
Average Inbound FS Size	Indicates the average number of bytes in the `COMMAREA` for inbound function ship requests. See "Average Inbound FS Size" on page 87
Average Outbound FS Size	Indicates the average number of bytes in the `COMMAREA` for outbound function ship requests. See "Average Outbound FS Size" on page 88.
Buffer Size	Reserved for future use.
Current Users	Indicates the number of current users of the file.
Disk File Name	Identifies the physical file on disk. The name in the Name datapoint is the dataset name, which is the logical file name. A single physical file may be associated with multiple dataset names.
Environment	Identifies the environment variable specifying the directory where the file is located.

TABLE 10-2 Files Window Datapoints *(Continued)*

Datapoint	Description
Flags	Indicates the file status as defined in the Sun MTP internal file control block structure. See "Flags" on page 93.
Group	Indicates the group to which the dataset is assigned.
Inbound FS	Indicates the total number of inbound function ship requests for this dataset. See "Inbound FS" on page 88.
Index Reads	Indicates the total number of index read I/O requests for this dataset. See "Index Reads" on page 89.
Index Writes	Indicates the total number of index write I/O requests for this dataset. See "Index Writes" on page 90.
Locks	Indicates the total number of locks obtained against this dataset. See "Locks" on page 91.
Outbound FS	Indicates the total number of outbound function ship requests for this dataset. See "Outbound FS" on page 91.
Reads	Indicates the total number of read I/O requests for this dataset. See "Reads" on page 92.
Remote System ID	Identifies the remote system hosting this file for function shipping.
Type	Indicates the file type. See "Type" on page 93.
Unlocks	Indicates the total number of locks released against this dataset. See "Unlocks" on page 94.
Waits	Indicates the total number of times application programs have been forced to wait for a record within this file. See "Waits" on page 94.
Writes	Indicates the total number of write I/O requests for this dataset. See "Writes" on page 95

Name

Description

The Name datapoint on the Files pane contains the name of the dataset.

Values

The dataset name is one to eight characters long.

Interpretation

An application program uses the dataset name to reference a particular file. The dataset name is mapped to a system file name (Disk File Name datapoint) whose location is defined in the Environment datapoint.

Alternate Index

Description

The Alternate Index datapoint indicates if the dataset is an alternate index file to a primary dataset.

Values

Value	Description
`true`	Indicates that the file is an alternate index file.
`false`	Indicates that the file is a primary dataset.

Interpretation

An alternate index accesses the same segments as the primary index because both actually reference the same data file (and its segments). An alternate index is defined in both the file table and the VSAM catalog. To create an alternate index, you must recreate the VSAM file with the Record Editor or the `unikixbld` utility, which automatically create alternates from the information configured in the file table and VSAM catalog.

Average Inbound FS Size

Description

The Average Inbound FS Size datapoint indicates the average number of bytes in the `COMMAREA` for inbound function ship (FS) requests for a dataset.

Values

An integer character count.

Interpretation

The Average Inbound FS Size datapoint is calculated by dividing the total number of characters received by inbound FS requests for a specific dataset by the number of requests (Inbound FS datapoint). Note that remote applications can send an FS request for a dataset defined as either local or remote to the region. In the latter case, the region issues an FS request of its own out to the remote system on behalf of the original requestor.

This datapoint provides a general idea of the volume of data transferred to and from a dataset defined on the local region due to FS requests from remote regions.

Average Outbound FS Size

Description

The Average Outbound FS Size datapoint indicates the average number of bytes in the COMMAREA for outbound function ship (FS) requests to a remote dataset.

Values

An integer character count.

Interpretation

The Average Outbound FS Size datapoint is calculated by dividing the total number of characters sent by outbound FS requests to a specific remote dataset by the actual number of requests (Outbound FS datapoint). Note that the administration tool only calculates Average Outbound FS Size and Outbound FS for a remote dataset.

This datapoint provides a general idea of the volume of data transferred to a particular remote dataset.

Inbound FS

Description

The Inbound FS datapoint indicates the number of inbound function ship (FS) requests that have occurred for a specific dataset.

Values

The value is a positive integer count.

Interpretation

An application program on a remote region can issue commands against a dataset defined on the local region. This remote request is considered an FS operation. Each request, no matter from which remote region, causes the Inbound FS count to increase. The remote application can request access to either a local or remote dataset. If the requested dataset is located on a remote system, Sun MTP function ships the request out to the appropriate remote system.

This datapoint helps identify the amount of traffic coming from remote regions to the local region for a particular dataset.

Index Reads

Description

The number of Index Reads represents the total number of physical reads issued to the index file since the region was started. A physical read is a read not accommodated by Sun MTP's internal buffers. This read may, however, be handled by the operating systems buffer cache and not actually require a physical disk access.

Values

Index Reads is a numeric quantity that is set to zero at region startup and increments for each physical read to the index portion of a dataset.

This datapoint is not set for remote datasets (those with a value set for the Remote System datapoint) because the administration tool does not communicate with the remote host.

Interpretation

This datapoint, together with Reads, Writes, and Index Writes, provides an accurate picture of dataset activity. Multiple highly active datasets on the same physical device decreases system throughput. All four of these datapoints should be used to determine the extent of a file's activity. Note that the Writes and Index Writes entail more overhead than their read counterparts.

There are ways to optimize dataset access performance. There are two kinds of configurable cache involved with a VSAM file local to a host machine: the operating system buffer cache and the Sun MTP buffer pool. The size of the Sun MTP buffer pool is easily configured in the VCT. Change the number of shared buffers, save the VCT to disk, and restart the region to activate the change.

There are also ways to distribute dataset files across hardware devices to achieve performance gains. The simplest way is to physically place entire datasets on separate disks. If the disks have separate disk controllers, performance can be further improved.

Another method to take advantage of I/O bandwidth is to use disk striping. This file system facility automatically distributes files across multiple devices. Again, consult your operating system's system administrator's guide for details.

Sun MTP provides another method to boost performance by spanning a dataset across multiple segments, each of which can reside on a separate device. Sun MTP allows a dataset to have as many as 8 segments. This method proves to be more efficient than the other two. You can choose to combine Sun MTP segmentation with operating system striping, but the results are not predictable since the file organization becomes extremely complex.

Index Writes

Description

The number of Index Writes represents the total number of physical index writes issued to the index file since the region was started. A physical write is a write not accommodated by Sun MTP's internal buffers. This write may, however, be handled by the operating systems buffer cache and not actually require a physical disk access.

Values

Index Writes is a numeric quantity that is set to zero when a region is started and increments for each physical write to the index portion of a dataset.

This datapoint is not set for remote datasets (those with a value set in the Remote System datapoint) because the administration tool does not communicate with a remote machine.

Interpretation

This datapoint, together with Writes, Reads, and Index Reads, provides an accurate picture of dataset activity. Multiple highly active datasets on the same physical device decreases system throughput. All four of these datapoints should be used to determine the extent of a file's activity. Note that the Writes and Index Writes entail more overhead than their read counterparts.

There are a few ways to optimize dataset access performance. There are two kinds of configurable cache involved with a VSAM file local to a host machine: the operating system buffer cache and the Sun MTP buffer pool. The size of the Sun MTP cache is easily configured in the VCT. Change the number of shared buffers, save the VCT to disk, and restart the region to activate a change.

There are also ways to distribute dataset files across hardware devices to achieve performance gains. The simplest way is to physically place entire datasets on separate disks. If the disks have separate disk controllers, performance can be further improved.

Another method to take advantage of I/O bandwidth is to use disk striping. This file system facility automatically distributes files across multiple devices. Again, consult your system's system administrator's guide for details.

Sun MTP provides another method to boost performance by spanning a dataset across multiple segments, each of which can reside on a separate device. Sun MTP allows a dataset to have any many as eight segments. This method proves to be more efficient than the other two. You can choose to combine Sun MTP segmentation with operating system striping, but the results are not predictable since the file organization becomes extremely complex.

Locks

Description

The Locks datapoint shows the total number of times a file required locking to process data on a particular dataset. There are two kinds of file locks: record locks and control interval (CI) locks. The Locks datapoint counts only CI locks. Because a CI can contain multiple records, or even portions of records, record locking is implicit in a CI lock. A CI lock occurs when an application program requests a record (contained inside a CI) from a file, whether for read or for write.

Values

The value of Locks ranges from 0 to *n*, where *n* depends on how frequently application programs lock a dataset.

Interpretation

This datapoint helps determine how heavily a file is used. For example, if the lock count is high and the transactions that access the file tend to run slowly, it is possible that transactions are waiting on each other due to CI resource contention. This could indicate a database design problem, or an application program problem in how, or in what sequence, the data is accessed.

The number of Locks and Unlocks should always be the same when no transactions are executing. If this is not the case, an internal Sun MTP problem has occurred. The system should be shut down and restarted. If this situation repeats itself, contact your authorized Sun service provider.

Outbound FS

Description

The Outbound FS datapoint shows the number of outbound function ships (FS) for a specific remote dataset.

Values

The value is an integer count.

Interpretation

Sun MTP sends an outbound FS request out under two conditions:

- A local application program issues a CICS command against a remote dataset. Sun MTP function ships the request to the system identified by the Remote System datapoint.
- A remote application program function ships a request for a dataset defined on the local region as remote. In this case, Sun MTP forwards the request by means of an outbound FS to the remote system.

Regardless of the situation, Sun MTP increments Outbound FS each time it processes an outbound FS request for access to a remote dataset.

Use this datapoint to monitor the traffic to a remote dataset.

Reads

Description

The number of Reads represents the total number of physical reads issued to a dataset since the region was started. A physical read is a read not accommodated by Sun MTP's internal buffers. This read may, however, be handled by the operating system buffer cache and not actually require a physical disk access.

Values

The Reads datapoint is an integer number that starts at zero when a region starts and increments for each physical read from (for KSDS, the data portion of) a dataset. Note that this datapoint is not set for remote datasets (those with a value set for the Remote System datapoint) because the administration tool does not communicate with the remote region.

Interpretation

This datapoint, together with Index Reads, Writes, and Index Writes, provides an accurate picture of dataset activity. Having multiple highly active datasets on the same physical device decreases system throughput. All four of these datapoints should be used to determine the extent of a dataset's activity. Note that the Writes and Index Writes entail more overhead than their read counterparts.

There are ways to optimize dataset access performance. There are two kinds of configurable cache involved with a VSAM file local to a host machine: the operating system buffer cache and the Sun MTP buffer pool. The size of the Sun MTP cache is easily configured in the VCT. Change the number of shared buffers, save the VCT to the disk, and restart the region to activate the change.

There are also ways to distribute dataset files across hardware devices to achieve performance gains. The simplest way is to physically place entire datasets on separate disks. If the disks have separate disk controllers, performance can be further improved.

Another method to take advantage of I/O bandwidth is to use disk striping. This file system facility automatically distributes files across multiple devices. Refer to your operating system's system administrator's guide for details.

Sun MTP provides another method to boost performance by spanning a dataset across multiple segments, each of which can reside on a separate device. Sun MTP allows a dataset to have as many as 8 segments. This method proves to be more

efficient than the other two. You can choose to combine Sun MTP segmentation with operating system striping, but the results are not predictable since the file organization becomes extremely complex.

Flags

Description

The Flags datapoint shows the file status as defined in the Sun MTP internal file control block structure.

Values

The flags are a series of bits that reflect the status of the file to Sun MTP.

This datapoint is internal to Sun MTP.

Type

Description

Type shows the dataset type, or database organization.

Values

The dataset Type indicates the file organization. Each of these types of database files are managed by the VSAM access method. The available types are:

- `KSDS`: Key-sequenced dataset
- `ESDS`: Entry-sequenced dataset
- `RRDS`: Relative record dataset

Interpretation

A KSDS file requires a key for access. The keys are kept in an index file. Thus, a KSDS dataset is broken into two files, data and index. Sun MTP uses the logical file name for a dataset, defined in the file table, to identify these files, with a `.dta` and `.idx` extension for the data and index files, respectively.

The ESDS and RRDS file organizations do not require an index file. Therefore, the file name defined in the file table corresponds to the actual system file name.

Verify that a dataset's type is consistent with respect to the application programs accessing a particular dataset. Build a particular type of dataset with the Sun MTP Record Editor.

Unlocks

Description

The Unlocks datapoint shows the total number of times a dataset was unlocked from a previous lock.

Values

The number of Unlocks is an integer that is initialized to zero at region startup. The value of the Unlocks datapoint increases as data activity takes place.

Interpretation

If Unlocks does not match Locks, then the dataset may either be locked for normal processing or possibly involved in a program logic error. An error or deadlock is evident when the two datapoints differ in value and do not change for an extended period of time.

When system performance seems to be degrading, examine the Unlocks and Locks datapoints for a system's datasets. If these values remain constant when they should both be increasing with file activity, the system is probably in deadlock. Note that these values should become identical and not change when the system is idle. If the values do not change and they are different, there may be an internal Sun MTP problem.

Waits

Description

The Waits datapoint represents the number of times application programs have to wait for a record of a particular dataset because another application program has already obtained the record.

Values

Sun MTP initializes the number of Waits to zero at region startup, and increments the number upon every record contention for a particular dataset.

Interpretation

A large number of Waits indicates that many user transactions are accessing the same data for a particular dataset. This could indicate a possible database design problem.

When application programs must wait on a resource, such as a record in a dataset, the entire application's performance degrades. If the application's performance appears to slump after reaching a certain transaction rate, analyze the Waits value to see if the number is increasing. If so, there might be a database design flaw or a bad mix of application programs might be running at the same time. They might all be trying to access the same records in the same dataset.

Writes

Description

The number of Writes represents the total number of physical writes issued to a data file since the region was started. A physical write is a write not accommodated by Sun MTP's internal buffers. This write might, however, be handled by the operating system buffer cache and not actually require a physical disk access.

Values

The Writes datapoint is an integer number that starts at zero when a region starts and increments for each physical write to (for KSDS, the data portion of) a dataset. This datapoint is not set for remote datasets (those with a value set for the Remote System datapoint) because the administration tool does not communicate with the remote region.

Interpretation

This datapoint, together with Index Writes, Reads, and Index Reads, provides an accurate picture of dataset activity. Having multiple highly active datasets on the same physical device decreases system throughput. All four of these datapoints should be used to determine the extent of a dataset's activity. Note that the Writes and Index Writes entail more overhead than their read counterparts.

There are ways to optimize dataset access performance. There are two kinds of configurable cache involved with a VSAM file local to a host machine: the operating system buffer cache and the Sun MTP buffer pool. The size of the Sun MTP cache is easily configured in the VCT. Change the number of shared buffers, save the VCT to the disk, and restart the region to activate the change.

Groups

The Groups window displays the following information about the groups defined for the region.

TABLE 10-3 Groups Window Datapoints

Datapoints	Description
Name	Identifies the group. Group names can be one to eight characters long.
Directory	Identifies the full path name of the directory where the group's table files are located. Refer to the *Sun Mainframe Transaction Processing Software Reference Guide* for information about grouping resources.
Read-Only	Indicates if the group directory is read-only. Values are `true` and `false`.
Sequence Number	Indicates the order in which Sun MTP loads groups.

Journals

The Journals window displays information about the Sun MTP journals. The following datapoints are displayed on the Journals window:

TABLE 10-4 Journals Window Datapoints

Datapoint	Description
Identifier	Journal file identifier. See "Identifier" on page 97.
Accounting	Indicates if this journal is used for accounting. Values are `true` and `false`. See "Accounting" on page 97.
Buffer Size	Indicates the buffer size in bytes for this journal. See "Buffer Size" on page 98.
Cached	Indicates whether the system cache is enabled for I/O to this journal. Values are `true` and `false`.
Environment	Identifies the environment variable that specifies the directory where the journal is located.

TABLE 10-4 Journals Window Datapoints *(Continued)*

Datapoint	Description
File Name	Identifies the file name associated with the journal.
File Size	Indicates the maximum size of this journal in bytes.
Group	Identifies the group to which the journal is assigned.

Identifier

Description

The Identifier datapoint is the file identifier assigned to this journal. Sun MTP can support up to 99 journals per region.

Values

Each journal is assigned a file identifier ranging from 01 to 99.

Interpretation

This file identifier is referenced by transaction and user resources when they want to enable accounting at a region and designate a particular journal file.

Accounting

Description

The Accounting datapoint indicates whether this journal is used for accounting purposes.

Values

Value	Description
`true`	Indicates that this is an accounting journal.
`false`	Indicates that this is not an accounting journal.

Interpretation

Accounting journals are archived by Sun MTP whenever they fill up or are closed. Refer to the *Sun Mainframe Transaction Processing Software Administrator's Guide* for information about accounting.

Buffer Size

Description

The Buffer Size datapoint indicates the size, in bytes, of the buffer for this journal.

Values

Numeric value ranging from 72 bytes to 32 KBytes.

Interpretation

Larger buffer sizes increase buffering and reduce the file I/O, thereby improving performance.

Mapsets

The Mapsets window displays information about the mapsets being used in the region.

The following datapoints are displayed on the Mapsets window:

TABLE 10-5 Mapsets Window Datapoints

Datapoint	Description
Name	Identifies the mapset; one to seven characters in length.
Current Users	Indicates the number of transactions that are currently using the mapset.
Disabled	Indicates if the mapset is disabled. See "Disabled" on page 99.
Group	Identifies the group to which mapset is assigned.
Loaded	Indicates if the mapset is loaded into shared memory. See "Loaded" on page 99
Executions	Indicates the total number of times this mapset has been executed. See "Executions" on page 100.

Disabled

Description

Indicates whether the mapset is currently disabled for use.

Values

Value	Description
`true`	Indicates that the mapset is disabled.
`false`	Indicates that the mapset is enabled.

Interpretation

If the mapset is disabled, requests to use the mapset will be denied and a response code returned to the requesting program.

Loaded

Description

The Loaded datapoint indicates whether the mapset is currently loaded into the region's memory.

Values

Value	Description
`true`	Indicates that the mapset is loaded.
`false`	Indicates that the mapset is not loaded.

Interpretation

If this datapoint is `true`, the mapset has been loaded into the region's memory and is available for use by applications.

If this datapoint is `false`, the next time this mapset is requested by a program, it will be loaded from the disk into memory.

Executions

Description

The Executions datapoint indicates the total number of times this mapset has been executed.

Values

An integer value initialized to zero at region startup and incremented each time an application program requests this mapset.

Interpretation

This datapoint can help identify high-use mapsets.

Programs

The Programs window shows information about the programs executing in the region. It displays information about every application program specified in the region's program table.

The Programs window helps the administrator determine which application programs are high use. These are candidates for optimization.

The following datapoints are displayed on the Programs pane:

TABLE 10-6 Programs Window Datapoints

Datapoint	Description
Name	Identifies the program. See "Name" on page 101.
API	Indicates the level of application programming interface (API) support. See "API" on page 102.
Aborts	Indicates the total number of times the program has aborted. See "Aborts" on page 102.
CPU	Indicates the total processor time (CPU) used by this program.
Current Users	Indicates the current number of transactions using this program.
DPL Inbound	Indicates the total number of times this program was invoked by a Distributed Program Link (DPL). See "DPL Inbound" on page 103.
DPL Outbound	Indicates the total number of times this program issued a DPL request. See "DPL Outbound" on page 103.
Disabled	Indicates if the program is disabled. See "Disabled" on page 104.

TABLE 10-6 Programs Window Datapoints *(Continued)*

Datapoint	Description
Executions	indicates the number of times program was executed. See "Executions" on page 104.
Group	Identifies the group to which this program is assigned.
Average Inbound CA Size	Indicates the average size of COMMAREA for inbound DPL invocations. See "Average Inbound CA Size" on page 105.
Java Class	Identifies the fully qualified Java™ class name. See "Java Class" on page 105
Language	Identifies the programming language this program is written in. See "Language" on page 106.
Loaded	Indicates whether the program is loaded into shared memory. See "Loaded" on page 106.
Average Outbound CA Size	Indicates the average size of COMMAREA for outbound DPL invocations. See "Average Outbound CA Size" on page 107.
Preload	Indicates if the program is preloaded. See "Preload" on page 107.
Remote Program	Identifies the program on remote system. See "Remote Program" on page 108.
Remote System ID	Four-character identifier of the remote system where the program is to execute. See "Remote System" on page 108.
Shared Library	Identifies the shared library associated with this program. See "Shared Library" on page 109.

Name

Description

The Name datapoint contains the name of a system executable program.

Values

The name can have a maximum of eight characters.

Interpretation

In order for Sun MTP to execute an application program, two conditions must be met. First, the program must be registered in the program table. Second, the executable must actually exist.

API

Description

The API datapoint indicates which API set the application program supports.

Values

The values of API are:

Value	Description
`D`	DPL API support
`F`	Full CICS API support

Interpretation

An application program that supports the full CICS API set can execute any CICS command. An application program that supports the DPL API set can execute only the API set corresponding to the DPL API set.

Programs that are called via the `EXEC CICS LINK` command and reside on a remote system are considered DPL and can only execute a subset of the full CICS API. For example, a program that resides on a remote system cannot execute an `EXEC CICS SEND` or `RECEIVE MAP` command because there is no terminal associated with the program on the remote system.

Aborts

Description

The Aborts datapoint shows the number of times a program has aborted since the region was started.

Values

The Aborts datapoint is a integer value that is initialized to zero when the region starts and is incremented each time a transaction aborts.

Interpretation

A program can abort for many reasons. For example:

- The program does not handle a CICS `HANDLE CONDITION` properly.
- A transaction server cannot recover from an application request (for example, divide by zero).
- A deadlock condition occurs.
- Sun MTP must make space in the recovery file.

A high number of aborts might indicate that transactions are deadlocking trying to obtain the same records in a database. The Transactions Aborted datapoint on the System Performance Overview pane represents the total number of aborts for all programs executing within the region.

DPL Inbound

Description

The DPL Inbound datapoint shows the total number of times this program was invoked by DPL requests from a remote region. These DPL requests can originate from any other region in the network configured in the system entries portion of the terminal table.

Values

The value of DPL Inbound is an integer value.

Interpretation

An application program on a remote region can issue a `LINK` command and specify a program that is located in a Sun MTP region. When the remote application program issues the `LINK`, the program name is looked up in the remote program table. If it is identified as a remote program, a DPL request is sent to the remote region, which can be a Sun MTP region.

DPL Outbound

Description

The DPL Outbound datapoint shows the total number of times this program issued a DPL request to remote regions.

Values

The value of DPL Outbound is an integer value.

Interpretation

A local application program can issue a `LINK` command and specify a program that resides on a remote region. When the program issues the `LINK`, the program name argument is looked up in the program table. If it is identified as a remote program, a DPL request is sent to the remote region.

Disabled

Description

The Disabled datapoint indicates whether the program is currently disabled for use.

Values

Value	Description
`true`	Indicates that the program is disabled
`false`	Indicates that the program is enabled

Interpretation

If the program is disabled, requests to use it are denied, and a response code is returned to the requesting program.

Executions

Description

The Executions datapoint indicates the total number of times the program was executed since the region was started.

Values

The Executions datapoint is an integer value initialized to zero at region startup and incremented each time an application program executes.

Interpretation

The Executions value is the sum of executions due to the following request types:

- An external request for a transaction by an end user
- Internal requests from another COBOL program
- An internal request due to a trigger condition for a transient data queue

This datapoint helps identify high use programs. The application programs that execute the most often should be targets for optimization to increase overall system performance.

Average Inbound CA Size

Description

The Average Inbound CA Size datapoint represents the average number of characters associated with each inbound DPL request for a particular program.

Values

The value is an integer representing a character count.

Interpretation

The average number of characters for inbound DPL requests is calculated by dividing the total number of characters that have been received by the number of inbound requests, DPL Inbound.

This datapoint provides a general idea of the volume of data transferred from a remote region to the local region for a specific program.

Java Class

Description

The Java Class datapoint defines the fully qualified Java class name for a Java program.

Values

The Java class name can contain up to 255 characters.

Interpretation

This is the name of the Java class that will be loaded by the Java virtual machine (JVM™) when Sun MTP attempts to load this program.

Note – The location of Java programs must be defined in the `KIXPROGS` environment variable. Refer to the *Sun Mainframe Transaction Processing Software Developer's Guide* for information about defining Java programs.

Language

Description

The Language datapoint indicates the language in which the program was written.

Values

TABLE 10-7 Language Types

Language	Description
`Batch`	A batch program started by Sun MTP.
`C`	C language
`COBOL`	COBOL
`Java`	Java
`PL/I`	PL/I
`System`	This is a Sun MTP program. The value `System` is used to differentiate it from user programs.
`Table`	An application data table that can be the target of an `EXEC CICS LOAD` request.

Loaded

Description

The Loaded datapoint indicates whether this program is load into the region's memory.

Values

Value	Description
`true`	Indicates that the program is loaded in memory
`false`	Indicates that the program is not loaded in memory

Interpretation

This is an indication that Sun MTP shared memory structures exist for this program.

Average Outbound CA Size

Description

The Average Outbound CA Size datapoint represents the average number of characters associated with each outbound DPL request made by a particular program.

Values

The Average Outbound CA Size datapoint is an integer value representing a character count.

Interpretation

The average number of characters for outbound DPL requests is calculated by dividing the total number of characters that have been sent to a remote region by the number of outbound requests, DPL Outbound.

This datapoint provides a general idea of the volume of data transferred from the local region to any remote region for a specific program.

Preload

Description

The Preload datapoint indicates whether this program is preloaded during region startup.

Values

Value	Description
`yes`	Program is to be preloaded
`no`	Program is not preloaded

Interpretation

This datapoint applies to shared library programs. If it is `yes`, the region attaches the program library during startup processing, thereby making it readily available for use.

Remote Program

Description

The Remote Program datapoint corresponds to the program name of a system executable at the remote region. Remote Program contains a value only if the local and remote names are different. By default, Sun MTP assumes the local and remote program names are the same. Refer to the *Sun Mainframe Transaction Processing Software Administrator's Guide* for information about using remote programs.

Values

The Remote Program name can have a maximum of eight characters. It cannot specify a mapset name.

Interpretation

In order for Sun MTP to execute a remote application program, the same conditions as for a local application program execution must hold true. First, the remote program must be registered in the program table. Second, the program must actually exist in the remote region.

Remote System

Description

The Remote System datapoint corresponds to the remote region where the program is deployed.

Values

The value of Remote System is a four-character identifier.

Interpretation

When a program in the program table is defined as remote, Sun MTP requests the remote region to initiate the DPL request. If undefined, Sun MTP uses a default mirror transaction identifier. The mirror transaction essentially performs the duties required to execute Program (or Remote Program if different from Program) on a remote region.

Shared Library

Description

The Shared Library datapoint specifies a shared object library that contains the system executable for the program identified in the Program datapoint.

Values

A Shared Library value can have a maximum of sixteen characters. It can include a path name prefix. The `.so` extension associated with shared object libraries is excluded from the Shared Library value.

Interpretation

The shared object library is located in a directory relative to the `$KIXSYS` value, unless the `KIXLIB` environment variable is set to a valid path name.

Terminals

The Terminals window shows the current status of all terminals connected to a region, as well as unconnected terminals defined in the Sun MTP terminal table. There are many types of connections: 3270, EPI, free, local, remote, TN3270, TR, and unknown.

- 3270 connections can be either real 3270 terminals or some form of 3270 emulation (Sun MTP treats them the same).
- An EPI connection is a programmatic connection via the external presentation interface API.
- Free entries refer to the number of terminal connections available to a region due to the licensed number of users. For example, a 16 user license automatically gives 16 free entries in the terminal table, and all entries that are not configured or connected appear as free.
- The local and remote connections refer to Sun MTP terminal handlers that run either on the host where the region runs, or remotely on another system, respectively.
- A TN3270 connection refers to a Telnet 3270 client. This type of 3270 emulation requires a TN3270 server (available with Sun MTP) that talks to a variety of TN3270 clients (many available in the public domain).
- The TR connection refers to a surrogate entry in the terminal table that handles transaction routing. TR is a special type of 3270 connection.
- In the case when the type of connection cannot be determined, the type is unknown.

The Terminals window can display a large number of datapoints. However, most datapoints are specific to the terminal type, and are only set if applicable.

The Terminals window displays the following datapoints:

TABLE 10-8 Terminals Window Datapoints

Datapoint	Description
Name	Terminal identifier.
Accounting Option	Indicates if accounting is on for this terminal.
Active	Indicates the current activity count for remotely connected terminals. See "Active" on page 111.
Busy	Indicates if a device is performing a transaction. See "Busy" on page 111.
Device	Identifies the terminal device name. See "Device" on page 112.
Group	Identifies the group to which this terminal is assigned.
In Service	Indicates if the terminal is in or out of service. See "In Service" on page 113.
LU Name	Identifies the logical unit name of the terminal handler. See "LU Name" on page 113.
Messages	Indicates the total number of messages for TCP/IP-based terminals. See "Messages" on page 114.
Model	Shows a specific 3270 model or the `TERM` environment variable. See "Model" on page 114.
PID	Process ID of the process handling this terminal. See "PID" on page 115.
Session ID	Shows the session identifier. See "Session ID" on page 116.
Status	Indicates the current status of a terminal. See "Status" on page 117.
Terminal Handler Host	Identifies the system where the terminal handler client process runs. See "Terminal Handler Host" on page 112.
Terminal Handler Port	Identifies the port number used by a terminal handler client process to communicate with a region. See "Terminal Handler Port" on page 116.
Terminal Handler Type	Identifies the type of terminal handler. See "Terminal Handler Type" on page 118.
User ID	Shows the user ID of the terminal user. See "User ID" on page 118.

Active

Description

The Active datapoint shows the polling activity for a socket-based terminal handler (TH) process that runs on behalf of a connected device. The polling activity shows if a device is alive or not. Sun MTP polls 3270 and remote connections (actually, the TH process) to see if it is still connected to the region. This datapoint is used for an entry with a type of 3270 or Remote.

Values

This value is zero if no connection exists. It alternates between the values 1 and 2 as polling occurs.

Interpretation

The Active datapoint is 0 when no polling is taking place. While a terminal is connected, the region pings the TH client process to determine if the process is still alive. After the region finds out that the TH process is alive, the Active datapoint takes a value of 2. Every subsequent polling interval, the region pings the TH client. If the TH does not respond within the next polling interval, the value for Active is decremented. If after two pings with no response, Active remains at 0, the region assumes the terminal has been disconnected. If a subsequent ping receives a response, Active is set back to 2.

If no response is received from a TH client process within the polling interval two times in a row (the value of Active reaches 0), Sun MTP assumes the device is no longer connected and frees up the terminal's entry in the region's terminal table (unless the entry is preconfigured). The datapoint immediately reflects any change in connectivity.

Busy

Description

The Busy datapoint indicates if a device is performing a transaction. This datapoint is used for all types of connected entries.

Values

The Busy datapoint displays either `true` or `false`. A value of `true` means that the device is performing a transaction.

Device

Description

The Device datapoint displays a form of device identifier based on the type of entry. This datapoint typically shows a terminal's definition as known to the client terminal handler process to help find the exact location of the terminal.

Values

The value of Device is dependent on the type of entry. The following table shows the kind of value to expect for a given terminal type:

Value	Description
EPI	An EPI-specific terminal identifier, at most eight characters.
Local	On a UNIX platform, this is the full path name of a physical device for a `tty` connection used by `unikixl` to identify the `tty`; usually of the form `/dev/tty*` with at most 20 characters displayed.
Remote	Same as Local, but on the remote system.
TN3270	TN3270-specific identifier, at most nine characters, used by the Sun MTP TN3270 server.
TR	The 12-character terminal name as known to the remote region; a concatenation of the Applid and the remote Termid.

Interpretation

The Device name helps identify the origin of a terminal table entry. This datapoint is used for all types of entries except for 3270 and free entries.

Terminal Handler Host

Description

The Host datapoint identifies the name of the system on which a terminal handler client process executes. A host name is displayed for all entry types (except Free and TR), since the value can be preconfigured in the terminal table.

Values

The Host datapoint displays the name of a host machine, up to 16 characters. If the name of a host cannot be determined, the datapoint displays `Unknown`.

Interpretation

The PID value for a terminal handler identifies a specific process running on a particular host. The Device Name for UNIX-based terminal handlers, Local and Remote, refers to a physical device name on the host.

The reason a host is not displayed for a TR type is that Sun MTP ignores the corresponding setting in the terminal table.

In Service

Description

The In Service datapoint indicates whether a 3270-like entry's device is in or out of service. A value appears for all (active or not) entries with a Type of 3270, EPI, TN3270, or TR.

Values

The In Service datapoint displays one of the following values:

Value	Description
`true`	Indicates that the device is in service
`false`	Indicates that the device is out of service

Interpretation

This datapoint indicates if a device is connected. The tool can show that inactive entries are in service if they are preconfigured in the terminal table. In this case, the In Service datapoint shows the value set in the table's In Svc datapoint.

When a terminal logs in to the region, it is put into service. An XPUx device can also be put into service with the `CEMT` transaction. For example:

```
CEMT SET TERMINAL L860 INSERVICE
```

The following `CEMT` transaction example takes a device out of service:

```
CEMT SET TERMINAL L860 OUTSERVICE
```

LU Name

Description

The LU Name datapoint displays the logical unit name of all 3270 entries in the Terminals window, as well as active transaction route (TR) entries.

Values

The LU Name datapoint displays an alphanumeric string that is always eight characters or less.

Interpretation

This datapoint is populated for all 3270 entries, active or not. This includes terminals preconfigured in the 3270 devices portion of the terminal table. For preconfigured terminals, the LU Name datapoint shows the value (of the same name) as set in the table. The Terminals window also shows the LU Name for active TR surrogate entries, since the underlying representation is essentially a 3270-type connection.

A preconfigured TR entry in the 3270 devices portion of the terminal table also sets the LU Name datapoint. However, the tool only displays the LU Name for active TR entries, since the actual value gets set upon connection. In other words, Sun MTP ignores the LU Name datapoint for preconfigured TR entries, and learns of the actual LU Name dynamically.

Messages

Description

The Messages datapoint displays the number of socket messages used to communicate back and forth between a remote terminal handler and a transaction server.

Values

The Messages datapoint is a cumulative, numeric value initialized to zero at region startup.

The tool computes the Messages datapoint only for remote and 3270 connections.

Interpretation

Sun MTP uses this number to ensure correct communication between the TH and a transaction server.

Model

Description

For most 3270-like entries, this datapoint shows the terminal model number and type. For other kinds of entries, the tool displays a context-dependent value for the type of terminal. The Model datapoint is only displayed for active entries.

Values

For 3270-like entries, this datapoint shows as much of the exact model as possible, for example, 3278 or 32782e. This is the case for connections with type equal to 3270 or EPI. TR connections merely show a generic value of 3270.

For the UNIX-based terminal handler `unikixl`, the Model datapoint shows the value of the `TERM` environment variable, for example, `xterms` or `vt100`, for an X terminal or a dumb terminal, respectively. TN3270 entries show a similar value, except the value represents the type of hardware the terminal handler runs on. For example, the Model is `xterm` for a TN3270 client.

Interpretation

The meaning of the Model datapoint is based on the type of the entry:

Type	Model
3270	Detailed Model number, if available
EPI	Detailed Model number, if available
Local	Value of `TERM` environment variable
Remote	Value of `TERM` environment variable
TN3270	Depends on where the TN3270 client runs
TR	Generic value of 3270

The Model datapoint can help identify additional terminal characteristics of a particular connection. This is especially true for 3270 and EPI type terminal handlers. For the UNIX-based terminal handlers, the Model datapoint tells if the end user is using an X terminal display or a dumb terminal. However, a user might set the `TERM` environment variable to a dumb terminal type, such as `vt100`, but still run the terminal handler in an `xterm`.

PID

Description

In all cases except one, the PID value is the system process ID of the process that handles a type of terminal window entry. The Host datapoint identifies the machine where the PID process is executing. The single exception is the PID for a transaction route entry, where there is no PID available to show.

Values

The PID datapoint is a nonzero integer, usually with a maximum of five digits. The operating system assigns the PID when a terminal handler process initiates execution.

Interpretation

The Terminals window shows the PID for terminal handlers that are actually connected to a region. Entries with a blank PID are either preconfigured in the terminal table or are free to be used by terminals that are auto-installed. The

Terminals window displays preconfigured entries of the terminal table to make the unconnected entries visible. For such entries, most datapoints are left blank because a connection does not exist.

Terminal Handler Port

Description

The Port datapoint shows the port number used by a terminal handler client process to communicate with a region. The Port is only used by socket-based clients.

Values

The Port datapoint is an integer value.

Interpretation

A region communicates with a terminal handler client through an operating system socket. A socket connects two ports, usually (but not necessarily) on different machines.

You cannot control the port number for the terminal handler client, because the operating system assigns them dynamically so that port numbers do not conflict within a host.

Note – The value in the Port datapoint is not the port number specified for the `unikixmain` server or the client initiator (`unikixi`).

Session ID

Description

A Session ID value is displayed for 3270, EPI, and TR entries. The Session ID datapoint is a number that identifies a particular connection between a device and some portion of the region or communication software on a machine that allows the device to communicate with a region.

Values

The Session ID datapoint has a nonzero integer value if a session exists. Otherwise, this datapoint displays nothing.

Interpretation

The interpretation of the Session ID datapoint depends on the type of entry. The following kinds of entries can display a value:

Type	Session Value
3270	The XPUx server dynamically assigns the session ID for the connection between itself and the terminal device
EPI	The value is used internally by Sun MTP to identify the connection
TR	The PU2.1 server dynamically assigns the session ID for the connection between itself and the transaction program.

Use the value to identify diagnostic entries in the trace files for a particular session.

Status

Description

The Status datapoint shows the current status of a terminal for any active entry.

Values

The Status datapoint can have the following values:

Value	Description
`P`	Indicates that a transaction is in progress
`E`	Shows that the terminal handler process is waiting for an Enter, in conversational mode, from the terminal to execute the transaction
Blank	Indicates that there is no activity for the terminal

Interpretation

The value for the Status datapoint is dependent upon the timing of the inquiry into the terminal's status. In other words, the Status datapoint shows the activity at the exact moment the agent obtains the status from Sun MTP. User activity constantly changes the status of a particular terminal.

Terminal Handler Type

Description

The Terminal Handler Type datapoint indicates the type of connection to a region rather than an actual device type. All connections to Sun MTP represent some form of 3270 communication. Each type of connection is handled by a context-specific terminal handler process or application programmatic interface (API). Therefore, type indicates the process type used to handle a particular terminal connection.

Values

The type can be one of the following values: 3270, EPI, free, local, remote, TN3270, TR, or unknown (if the type is truly not known by the agent). An asterisk (*) prefix might be displayed if there is something wrong with the entry. For example, all items of unknown type show the following value: `*Unknown`.

User ID

Description

The user ID associated with an entry in the Terminals window could either be a user ID from the Sun MTP Sign-On Table (SNT) or a system login user ID. An entry can be for a variety of terminal handler client and server processes.

Values

The User ID datapoint displays a user ID up to eight characters. The user ID is most often the system login user ID of the user involved in a terminal table entry. The login user ID is a value specified in the password file (usually `/etc/passwd` or a similar NIS map file).

Interpretation

There are several types of Sun MTP terminal handlers and server processes, and the User ID datapoint has a different meaning for each type of entry:

TH/Server Process	User ID
3270	User ID of the 3270 client initiator (`unikixi`)
EPI	User ID of the system process communicating to the region by means of the EPI protocol
Local	User ID running `unikixl` on the local machine
TN3270	User ID of the TN3270 server (`unikixtnemux`)
TR	User ID of the user on the remote region

The user ID for a 3270 terminal handler can be overridden by specifying a user ID to VTAM or XPUx when linking up to a Sun MTP region. This case and a transaction route's user ID (where the user ID comes from CICS/Sun MTP) are the only cases where the user ID is not an operating system user ID.

Temporary Storage Table

The TS Table window displays information about the configured Sun MTP temporary storage resources.

The TS Table window displays the following datapoints:

TABLE 10-9 Temporary Storage Table Window Datapoints

Datapoint	Description
FS Average Bytes Inbound	Indicates the average character count for inbound function ship requests. See "FS Average Bytes Inbound" on page 119.
FS Average Bytes Outbound	Indicates the average character count for outbound function ship requests. See "FS Average Bytes Outbound" on page 120.
Group	Identifies the group to which this entry is assigned.
FS Total Inbound	Indicates the total number of inbound function ship requests for a particular temporary storage queue.
FS Total Outbound	Indicates the total number of outbound function ship requests for a particular temporary storage queue.
Name	Identifies the temporary storage queue.
Recoverable	Indicates whether this temporary storage queue is recoverable. Values are `true` or `false`.
Remote TS Queue	Identifies this temporary storage queue as defined in the corresponding remote system defined by the Remote System ID.
Remote System ID	Identifies the remote system to which requests will be shipped.

FS Average Bytes Inbound

Description

This datapoint represents the average number of characters per inbound function ship (FS) request for a particular Temporary Storage (TS) queue.

Values

The value of the datapoint is an integer character count.

Interpretation

This datapoint is calculated by dividing the total number of characters received by inbound FS requests for a specific TS queue by the number of requests, FS Total Inbound. Note that remote applications can send an FS request for a TS queue defined as either local or remote to the Sun MTP region. For the latter case, the region issues an FS request of its own out to the remote system (for the remote queue) on behalf of the original requestor.

This datapoint provides a general idea of the volume of data transferred to and from a TS queue defined on the local Sun MTP region due to FS requests from (possibly multiple) remote Sun MTP and CICS regions.

FS Average Bytes Outbound

Description

This datapoint represents the average number of characters per outbound function ship (FS) request for a particular remote temporary storage (TS) queue.

Values

The value of the datapoint is an integer character count.

Interpretation

This datapoint is calculated by dividing the total number of characters sent by outbound FS requests to a specific remote TS queue by the number of requests, FS Total Outbound.

This datapoint provides a general idea of the volume of data transferred to a particular remote TS queue.

Temporary Storage Queues

The Temporary Storage (TS) Queues window displays information about active TS queues for a region. It displays the following datapoints:

TABLE 10-10 Temporary Storage Queues Window Datapoints

Datapoint	Description
Name	Identifies this TS queue.
Last Read	Identifies the record number of last record read from this TS queue.

TABLE 10-10 Temporary Storage Queues Window Datapoints

Datapoint	Description
Last Written	Identifies the record number of the last write to this TS queue.
Main	Indicates whether this TS queue is assigned to main storage or auxiliary (disk) storage. Values are `true` or `false`.
Size	Indicates the number of bytes currently used by this TS queue.

Extrapartition Transient Data Queues

The Extrapartition Transient Data Queue (TDQ) window displays information about extrapartition transient data (TD) queues.

It displays the following datapoints:

TABLE 10-11 Extrapartition TDQ Window Datapoints

Datapoint	Description
Name	Identifies the extrapartition TD queue. See "Name" on page 122.
Open	Indicates if the queue is open for access. See "Open" on page 125.
Enabled	Indicates if the queue is enabled. Values are `true` and `false`.
Environment	Identifies a system environment variable specified for this queue. See "Environment" on page 122.
Disk File Name	Identifies the file associated with this queue. See "Disk File Name" on page 123.
Format	Identifies the record format of the file See "Format" on page 124.
Group	Identifies the group to which this queue is assigned.
Input	Indicates if this is an input queue. Values are `true` and `false`.
Record Length	Identifies the length of a record to be sent to this queue. See "Record Length" on page 124.
FS Total Inbound	Indicates the number of inbound function ship requests for this queue. See "FS Total Inbound" on page 124.
Last Read Time	Indicates when the last record was read from this queue. See "Last Read Time" on page 125.
Last Write Time	Indicates when the last record was written to this queue. See "Last Write Time" on page 126.

TABLE 10-11 Extrapartition TDQ Window Datapoints *(Continued)*

Datapoint	Description
Last Read	Identifies the record number of last read.
Last Written	Identifies the record number of last written.
FS Average Bytes Inbound	Indicates the average character count for inbound function ship requests. See "FS Average Bytes Inbound" on page 126.

Name

Description

The Name datapoint is the name of a transient data (TD) queue defined in the Extrapartition Destinations portion of the Sun MTP Destination Control Table (DCT).

Values

The Name datapoint is four characters.

Interpretation

Application programs use the name of a TD queue to specify a particular queue within the application code.

Environment

Description

The Environment datapoint shows the name (not the value) of a system environment variable specified for a given TD queue in the Extrapartition Destinations section of the DCT.

Values

By convention, the name of an environment variable is an uppercase character string of at most 14 characters. The environment variable value contains a character string that specifies a file system directory. To view the value of the Environment variable, echo the value at the system prompt where the region was started, as follows:

```
$ echo $variable-name
```

where *variable-name* is the name displayed in the Environment datapoint.

Interpretation

The value of the Environment variable specifies the system directory that contains the file for a queue. The actual name of the queue appears in the Disk File Name datapoint. Concatenate the value of Disk File Name to the end of the value of the Environment for the full path name of the file that contains the TD queue.

Disk File Name

Description

The Disk File Name datapoint contains the name of the file associated with a TD queue.

Values

A standard system file name with a maximum of eight characters.

Interpretation

The Disk File Name identifies a variable-length KSDS VSAM file, with a maximum record size of 32 KBytes. Concatenate the value of Disk File Name to the end of the value of the Environment to get the full path name of the file used by the TD queue.

Format

Description

The Format datapoint indicates the record format of the file.

Values

Value	Description
LINE	Line file of character data; a line feed at the end of each line
JOB	Line file of character data, linked to kixjob shell script
PRINT	Line file of character data, linked to kixprint shell script
RECORD	Fixed-length records with character or binary data
RECORDV	Variable-length records with character or binary data

FS Total Inbound

Description

The FS Total Inbound datapoint shows the number of inbound FS requests for a particular TD queue.

Values

The value of the datapoint is an integer count of requests.

Interpretation

The FS Total Inbound datapoint shows the number of function ship requests that the local node has received from remote nodes requesting a particular TD queue.

This datapoint gives you a general idea of the volume of data that is being transferred from multiple remote nodes to the local node for a specific TD queue.

Record Length

Description

The Record Length datapoint displays the length, in bytes, of a record to be sent to a particular extrapartition TD queue.

Values

The value of the Record Length datapoint is an integer number of bytes in the record.

Open

Description

The Open datapoint indicates if a particular queue is open for access.

Values

Value	Description
`false`	Indicates that the TD queue is closed
`true`	Indicates that the TD queue is open (default)

Interpretation

This datapoint displays the current state of the TD queue. The value can be set for startup through the DCT Extrapartition Destinations screen, or dynamically with the `CEMT SET` command.

Last Read Time

Description

This datapoint indicates when the last record was read from a TD queue.

Values

The format is *ddd mmm nn hh:mm:ss*-GMT-*hh:mm yyyy*

where:

ddd: day
mmm: month
nn: date
hh: hours
mm: minutes
ss: seconds
yyyy: year

If there has never been an entry on a TD queue, this datapoint displays blanks

Interpretation

Sun MTP records the exact time when an application reads from a TD queue with a `READQ TD` command.

Last Write Time

Description

This datapoint indicates when the last record was written to a particular TD queue.

Values

The format is *ddd mmm nn hh:mm:ss*-GMT-*hh:mm yyyy.*

where:

ddd: day
mmm: month
nn: date
hh: hours
mm: minutes
ss: seconds
yyyy: year

If there has never been an entry on a TD queue, this datapoint displays blanks.

Interpretation

Sun MTP records the exact time when an application writes to a TD queue with a `WRITEQ TD` command.

FS Average Bytes Inbound

Description

The FS Average Bytes Inbound datapoint represents the average number of characters per inbound FS request for a particular TD queue.

Values

The value of this datapoint is an integer character count.

Interpretation

This datapoint is calculated by dividing the value the total number of characters received by inbound FS requests for a specific TD queue by the number of requests, FS Total Inbound. Note that remote applications can send an FS request for a TD queue defined as either local or remote to the region. For the latter case, the region issues an FS request of its own out to the remote system (for the remote queue) on behalf of the original requestor.

This datapoint provides a general idea of the volume of data transferred to and from a TD queue defined on the local Sun MTP region due to FS requests from (possibly multiple) remote regions.

Intrapartition Transient Data Queues

The Intrapartition Transient Data Queue (TDQ) pane displays information about intrapartition queues. It displays the following datapoints:

TABLE 10-12 Intrapartition TDQ Window Datapoints

Datapoint	Description
Name	Identifies the intrapartition TD queue. See "Name" on page 128.
Enabled	Identifies if the queue is enabled. Values are `true` and `false`.
Environment	Identifies a system environment variable specified for this queue. See "Environment" on page 128.
Facility	Indicates one of the following: `TERMINAL`, `FILE`, or `PRINTER`.
Disk File Name	Identifies the file associated with this queue. See "Disk File Name" on page 129.
Group	Identifies the group to which this queue is assigned.
Recoverable	Indicates if the queue is recoverable. Values are `true` and `false`.
Terminal ID	Value specified by the `TERMID` parameter in the `EXEC CICS` command. See "Terminal ID" on page 130.
FS Total Inbound	Indicates the number of inbound function ship requests for this queue. See "FS Total Inbound" on page 129.
Transaction ID	Identifies the transaction identifier of the transaction to execute when the number of entries in the queue reaches the Trigger Level value. See "Transaction ID" on page 131.
Trigger Level	Identifies a threshold value for the number of entries on a transient data (TD) queue. See "Trigger Level" on page 131.
Last Read Time	Indicates when the last record was read from this queue. See "Last Read Time" on page 130.
Last Write Time	Indicates when the last record was written to this queue. See "Last Write Time" on page 130.
Last Read	Identifies the record number of last read.
Last Written	Identifies the record number of last written.
FS Average Bytes Inbound	Indicates the average character count for inbound function ship requests. See "FS Average Bytes Inbound" on page 132.

Name

Description

The Name datapoint is the name of a TD queue defined as the DestID in the Intrapartition Destinations portion of the DCT.

Values

The Name datapoint consists of four characters.

Interpretation

Application programs use the name of a TD queue to specify a particular queue within the application code.

Environment

Description

The Environment datapoint shows the name (not the value) of a system environment variable specified for a given TD queue in the Intrapartition Destinations section of the DCT.

Values

By convention, the name of an environment variable is an uppercase character string of at most 14 characters. The environment variable value specifies a character string that specifies a file system directory. To view the value of the Environment variable, echo the value at the system prompt where the region was started, as follows:

```
$ echo $variable-name
```

where *variable-name* is the name displayed in the Environment datapoint.

Interpretation

The value of the Environment variable specifies the system directory that contains the file for a queue. The actual name of the queue appears in the Disk File Name datapoint. Concatenate the value of Disk File Name to the end of the value of the Environment for the full path name of the file that contains the TD queue.

Disk File Name

Description

The Disk File Name datapoint is the name of the file associated with a TD queue.

Values

A standard system file name with a maximum of 14 characters.

Interpretation

The Disk File Name datapoint identifies a variable-length KSDS VSAM file, with a maximum record size of 32 KBytes. Concatenate the value of Disk File Name to the end of the value of Environment to get the full path name of the file used by the TD queue.

FS Total Inbound

Description

The FS Total Inbound datapoint shows the number of inbound FS requests for a particular TD queue.

Values

The value of the datapoint is an integer count of requests.

Interpretation

The FS Total Inbound datapoint shows the number of function ship requests that the local node has received from remote nodes requesting a particular TD queue.

This datapoint gives you a general idea of the volume of data that is being transferred from multiple remote nodes to the local node for a specific TD queue.

Last Read Time

Description

This datapoint indicates the most recent time a record was read from a TD queue.

Values

The format is *ddd mmm nn hh:mm:ss*-GMT-*hh:mm yyyy.*

where:

ddd: day
mmm: month
nn: date
hh: hours
mm: minutes
ss: seconds
yyyy: year

If there has never been an entry on a TD queue, this datapoint displays blanks.

Interpretation

Sun MTP records the exact time when an application reads from a TD queue with a `READQ TD` command.

Last Write Time

Description

This datapoint indicates the most recent time a record was written to a particular TD queue.

Values

The format is *ddd mmm nn hh:mm:ss*-GMT-*hh:mm yyyy.* If there has never been an entry on a TD queue, this datapoint displays blanks.

Interpretation

Sun MTP records the exact time when an application writes to a TD queue with a `WRITEQ TD` command.

Terminal ID

Description

The Terminal ID datapoint represents the value specified by the `TERMID` parameter in the `EXEC CICS` command. It is the terminal identifier that the transaction is to be executed against.

Values

The value is any valid four-character name. If the command is to execute locally, The terminal ID must match an entry in the Sun MTP terminal table.

Interpretation

The Terminal ID datapoint specifies the name of a terminal where the task will run. If omitted, the task runs without a terminal. If a Terminal ID datapoint is not displayed, it indicates that the application program cannot perform I/O to a terminal.

Transaction ID

Description

The Transaction ID datapoint contains the transaction identifier of the transaction to execute when the number of entries in a TD queue reaches the Trigger Level value.

Values

The Transaction ID datapoint is a one- to four-character identifier specified in the Intrapartition Destinations portion of the Sun MTP DCT.

Interpretation

When a TD queue reaches its trigger value, the transaction specified by Transaction ID is invoked. Therefore, in order to be executed, the Transaction ID must have an entry in the transaction table.

Trigger Level

Description

The Trigger Level datapoint is a threshold value for the number of entries on a TD queue. When the TD queue reaches the trigger level, a specific transaction ID is executed.

Values

The Trigger Level datapoint is a numeric value between 0 and 99,999. The administrator defines the value for a particular TD queue in the Intrapartition Destinations portion of the DCT.

Interpretation

When a Trigger Level value is zero, Sun MTP never starts the transaction. When the number of entries on a TD queue reaches the Trigger Level value, Sun MTP starts the transaction specified by transaction ID. In order for Sun MTP to start the transaction, the transaction ID must have an entry in the transaction table.

FS Average Bytes Inbound

Description

The FS Average Bytes Inbound datapoint represents the average number of characters per inbound FS request for a particular transient data queue.

Values

The value of the datapoint is an integer character count.

Interpretation

This datapoint is calculated by dividing the value the total number of characters received by inbound FS requests for a specific TD queue by the number of requests, FS Total Inbound. Note that remote applications can send an FS request for a TD queue defined as either local or remote to the region. For the latter case, the region issues an FS request of its own out to the remote system (for the remote queue) on behalf of the original requestor.

This datapoint provides a general idea of the volume of data transferred to and from a TD queue defined on the local Sun MTP region due to FS requests from (possibly multiple) remote regions.

Remote Transient Data Queues

The Remote Transient Data Queue (TDQ) pane displays the following information about remote transient data queues:

TABLE 10-13 Remote TDQ Window Datapoints

Datapoint	Description
Name	Identifies the remote TDQ. See "Name" on page 133.
Enabled	Indicates if the queue is enabled. Values are `true` and `false`.
FS Average Bytes Inbound	Indicates the average size of inbound function shipping requests in bytes.
FS Average Bytes Outbound	Indicates the average character count for outbound function ship requests. See "FS Average Bytes Outbound" on page 133.
FS Total Inbound	Indicates the total number of inbound function shipping requests.
FS Total Outbound	Indicates the number of outbound function ship requests for this queue. See "FS Total Outbound" on page 134
Group	Identifies the group to which this queue is assigned.

TABLE 10-13 Remote TDQ Window Datapoints *(Continued)*

Datapoint	Description
Record Length	Indicates the length of the record to be sent to the remote TDQ. See "Record Length" on page 134.
Remote System ID	Identifies the remote system that hosts the TDQ.
Terminal ID	Identifies the terminal name used to run the triggered transaction.

Name

Description

The Name datapoint represents the name of the remote TD queue that an application program references in the `WRITEQ TD`, `DELETEQ TD`, and `READQ TD` commands. The application accesses Name over the Remote System ID connection.

Values

The Name datapoint is a one- to eight-character name. The name of a TD queue is defined by the application program when it creates a queue with the first `WRITEQ TD` command issued for a particular remote queue.

FS Average Bytes Outbound

Description

The FS Average Bytes Outbound datapoint represents the average number of characters associated with each outbound FS request for a TD queue.

Values

The value of the datapoint is an integer character count.

Interpretation

The average number of characters for outbound TD function ship requests is calculated by dividing the total number of characters that have been sent by the number of outbound TD function ship requests (FS Total Outbound).

This datapoint can give you a general idea of the volume of data that is being transferred to multiple remote nodes from the local node for the specified queue.

FS Total Outbound

Description

This datapoint shows the number of outbound FS requests for a given TD queue.

Values

The value of the datapoint is a number representing a request count.

Interpretation

Sun MTP processes an outbound FS request under two circumstances. First, a local application program can issue a `READQ TD` or `WRITEQ TD` against a remote TD queue. Sun MTP ships the request to the system identified by Remote System ID for the remote queue. The other condition where this happens is by a request generated by a remote application program. Such a program's region could have the TD queue defined as residing in the Sun MTP region while the TD queue can actually reside somewhere else. In this case, Sun MTP forwards the request to the Remote System ID that it knows about. Regardless of the situation, Sun MTP increments FS Total Outbound each time it processes a FS for access to a remote TD queue.

This datapoint can give you a general idea of the volume of data transferred to remote regions from the local Sun MTP region for a specific TD queue.

Record Length

Description

The Record Length datapoint contains the length of the record to be sent to a remote TD queue.

Values

The value is an integer number of bytes in the record.

Interpretation

The Record Length is specified in the TD queue definition in the Remote Destinations portion of the Sun MTP DCT.

Remote Systems

The Remote Systems pane displays the following information about remote systems:

TABLE 10-14 Remote Systems Window Datapoints

Datapoint	Description
Name	Identifies the remote system.
Access Method	Indicates the access method; either TCP/IP or SNA.
Access Method Parameter	Additional access method information.
AP Average Bytes In	Indicates the average bytes inbound for asynchronously started transactions.
AP Average Bytes Out	Indicates the average bytes outbound for asynchronously started transactions.
AP Total Bytes In	Indicates the total bytes inbound for asynchronously started transactions.
AP Total Bytes Out	Indicates the total bytes outbound for asynchronously started transactions.
AP Total Requests In	Indicates the total requests inbound for asynchronously started transactions.
AP Total Requests Out	Indicates the total requests outbound for asynchronously started transactions.
Average Bytes In	Indicates the average number of bytes inbound per ISC request.
Average Bytes Out	Indicates the average number of bytes outbound per ISC request.
DPL Average Bytes In	Indicates the average number of bytes inbound per Distributed Program Link (DPL) request.
DPL Average Bytes Out	Indicates the average number of bytes outbound per DPL request.
DPL Total Bytes In	Indicates the total number of bytes inbound for DPL requests.
DPL Total Bytes Out	Indicates the total number of bytes outbound for DPL requests.
DPL Total Requests In	Indicates the total number of inbound DPL requests.
DPL Total Requests Out	Indicates the total number of outbound DPL requests.
DTP Average Bytes In	Indicates the average number of bytes inbound per Distributed Transaction Processing (DTP) request.

TABLE 10-14 Remote Systems Window Datapoints *(Continued)*

Datapoint	Description
DTP Average Bytes Out	Indicates the average number of bytes outbound per DTP request.
DTP Total Bytes In	Indicates the total number of bytes inbound for DTP requests.
DTP Total Bytes Out	Indicates the total number of bytes outbound for DTP requests.
DTP Total Requests In	Indicates the total number of inbound DTP requests.
DTP Total Requests Out	Indicates the total number of outbound DTP requests.
FC Average Bytes In	Indicates the average number of bytes inbound per file control function shipping request.
FC Average Bytes Out	Indicates the average number of bytes outbound per file control function shipping request.
FC Total Bytes In	Indicates the total number of bytes inbound for file control function shipping requests.
FC Total Bytes Out	Indicates the total number of bytes outbound for file control function shipping requests.
FC Total Requests In	Indicates the total number of inbound file control function shipping requests.
FC Total Requests Out	Indicates the total number of outbound file control function shipping requests.
Host	Identifies the host name of the LU server.
In Service	Indicates if the remote system is in service.
Local LU Name	Identifies the LU name of the local system.
Mode Name	Identifies the mode name for session parameters.
Remote LU Name	Identifies the LU name of the remote system.
SYS Total Requests In	Indicates the total number of inbound system requests.
SYS Total Requests Out	Indicates the total number of outbound system requests.
TD Average Bytes In	Indicates the average number of bytes inbound per transient data function shipping request.
TD Average Bytes Out	Indicates the average number of bytes outbound per transient data function shipping request.
TD Total Bytes In	Indicates the total number of bytes inbound for transient data function shipping requests.

TABLE 10-14 Remote Systems Window Datapoints *(Continued)*

Datapoint	Description
TD Total Bytes Out	Indicates the total number of bytes outbound for transient data function shipping requests.
TD Total Requests In	Indicates the total number of inbound transient data function shipping requests.
TD Total Requests Out	Indicates the total number of outbound transient data function shipping requests.
TR Total Requests In	Indicates the total number of inbound transaction route requests.
TR Total Requests Out	Indicates the total number of outbound transaction route requests.
Trace	Indicates whether tracing is enabled for the system.
TS Average Bytes In	Indicates the average number of bytes inbound per temporary storage function shipping request.
TS Average Bytes Out	Indicates the average number of bytes outbound per temporary storage function shipping request.
TS Total Bytes In	Indicates the total number of bytes inbound per temporary storage function shipping request.
TS Total Bytes Out	Indicates the total number of bytes outbound per temporary storage function shipping request.
TS Total Requests In	Indicates the total number of inbound temporary storage function shipping requests.
TS Total Requests Out	Indicates the total number of outbound temporary storage function shipping requests.
Total Requests In	Indicates the total number of inbound requests for the system.
Total Requests Out	Indicates the total number of outbound requests for the system.

Transactions

The Transaction window displays information about the transactions specified in the transaction table. This information can be useful in determining high-use transactions, which are candidates for optimization. In particular, the Aborts datapoint shows that how often transactions are aborting. When this happens, check the Programs window to determine which program is aborting.

The following datapoints are displayed on the Transactions window:

TABLE 10-15 Transactions Window Datapoints

Datapoint	Description
Name	Transaction identifier. See "Name" on page 139.
Aborts	Indicates the total number of times this transaction aborted. See "Aborts" on page 140.
Accounting Journal	Accounting journal identifier.
Accounting Option	Indicates if accounting is on or off for the transaction. See "Accounting Option" on page 141.
Acct File ID	Identifies the assigned accounting journal. See "Acct File ID" on page 142
AP Average Bytes In	Indicates the average number of bytes per asynchronous processing (AP) request inbound. See "AP Average Bytes In" on page 142.
AP Average Bytes Out	Indicates the average number of bytes per AP request outbound. See "AP Average Bytes Out" on page 142.
APPC Enabled	Indicates whether the transaction is a back-end DTP transaction. See "APPC Enabled" on page 140.
CPU	Indicates the total processor usage. See "CPU" on page 143.
CPU (System)	Indicates the total system processor usage. See "CPU System" on page 143.
CPU (User)	Indicates the total user processor usage. See "CPU User" on page 144.
Disabled	Indicates if this transaction is disabled. See "Disabled" on page 144.
Dump Suppressed	Indicates if dump production is inhibited. See "Dump Suppressed" on page 144.
Executions	Indicates the total number of times the transaction has been executed. See "Executions" on page 145.
Group	Identifies the group to which the transaction is assigned.

TABLE 10-15 Transactions Window Datapoints *(Continued)*

Datapoint	Description
ISC AP Total Requests In	Indicates the total ISC AP requests (inbound). See "ISC AP Total Requests In" on page 145.
ISC AP Total Requests Out	Indicates the total ISC AP requests (outbound). See "ISC AP Total Requests Out" on page 146.
Local Queue	Indicates whether the transaction is locally queued if the remote system in unavailable when the transaction is shipped. If true, the transaction is queued locally. If false, the transaction is not queued locally.
Program	Identifies the first program invoked by this transaction. See "Program" on page 146.
Remote System ID	Identifies the remote system to which requests will be shipped.
Remote TranID	Identifier of the transaction on the remote system. See "Remote TranID" on page 147.
Screen Size	Indicates the screen size allowed for the transaction. See "Screen Size" on page 148.
Security Keys	Indicates the security level of this transaction. See "Security Keys" on page 149.
STARTs	Indicates the number of times this transaction was started internally. See "STARTs" on page 147.
TWA Size	Indicates the size in bytes of any transaction work area assigned to the transaction. Values range from 0 (no TWA) to a maximum of 32 KBytes.
Tx Class	Identifies the transaction class to which transaction is assigned. See "TX Class" on page 148.

Name

Description

The Name datapoint shows the transaction identifier as specified in the region's transaction table, which is used to map a transaction identifier to a program for execution.

Values

A four-character identifier.

Interpretation

A user or program runs an application program by requesting its transaction ID.

- The user types the transaction ID at a terminal session.
- An application issues a programmatic request with the transaction ID as a parameter.
- A transient data queue can also request a transaction ID when a trigger condition is met.

If a requested transaction is not defined in the transaction table, the user receives a message that the transaction ID is invalid.

APPC Enabled

Description

The APPC Enabled datapoint indicates whether a transaction is a back-end DTP transaction.

Values

Value	Description
`true`	Indicates that the transaction is a back-end DTP program and is capable of performing APPC functions.
`false`	Indicates that the transaction is not a back-end DTP program. It can, however, still perform outbound ISC functions.

Interpretation

The APPC Enabled datapoint value reflects the APPC datapoint value of an entry in the transaction table.

Aborts

Description

The Aborts datapoint shows the number of times a particular transaction has aborted since the region was started.

Values

The Aborts datapoint is a integer value initialized to zero when the region starts and incremented each time a transaction aborts.

Interpretation

A transaction can abort for many reasons. For example, the application program does not handle a CICS `HANDLE CONDITION` properly, or a transaction server cannot recover from an application request (for example, divide by zero), or a deadlock condition occurs, or the region aborts a transaction to make space in the recovery file.

A high incidence of transaction aborts can indicate that many transactions are deadlocking trying to obtain the same records in a database. Confirm this by examining the Transactions Deadlocked datapoint on the Performance Overview window. The Transactions Aborted datapoint on the Performance Overview window represents the total number of aborts for all transactions executing within the region. When these two values increase together, you might have to review the database access protocol. Check the Programs window to identify the programs causing the aborts.

Accounting Option

Description

The Accounting Option datapoint indicates whether accounting is specified for the transaction.

Values

The Accounting Option datapoint has one of the following values:

Value	Description
`Y`	Accounting is on for this transaction
`N`	Accounting is off for this transaction
`D`	Accounting is controlled by the transaction accounting setting in the Monitoring Control Table (MCT)

Interpretation

The value of `Y` for the Accounting Option datapoint does not necessarily mean that the region generates accounting records for a particular transaction. The same condition applies when the Accounting Option datapoint is `D`. Accounting must be specified at both the transaction and global levels in order for the region to generate accounting records for a transaction.

Acct File ID

Description

The Acct File ID datapoint displays the number of the journal that receives accounting records for this transaction.

Values

An integer number between 1 and 99.

This datapoint displays a zero if accounting is not specified for the transaction.

Interpretation

The administrator can exert a fair degree of control over the target locations for accounting records. For example, each transaction could theoretically have its own accounting record file. The administrator can also group the accounting records of some number transactions into a single shared file.

AP Average Bytes In

Description

The AP Average Bytes In datapoint shows the average number of characters associated with each inbound ISC asynchronous processing (AP) request for a particular transaction from a remote region.

Values

The value is a character count.

Interpretation

The value is calculated by dividing the total number of characters that have been received from a remote region by the number of inbound requests (ISC AP Total Requests In).

This datapoint gives a general idea of the volume of data for a particular transaction that is being transferred to the local region from a remote region.

AP Average Bytes Out

Description

The AP Average Bytes Out datapoint shows the average number of characters associated with each outbound ISC asynchronous processing (AP) request for a particular transaction at a remote region.

Values

The value is a character count.

Interpretation

The value is calculated by dividing the total number of characters that have been sent to the remote region by the number of outbound requests, (ISC AP Total Requests Out).

This datapoint gives a general idea of the volume of data for a particular transaction that is being transferred from the local region to a remote region.

CPU

Description

The CPU datapoint shows the accumulated amount of CPU time spent executing a transaction since the region was started.

Values

The value is displayed as the total number of seconds.

Interpretation

The CPU value of a transaction includes both kernel (system call) and user (application code) execution time. If a region uses a database access method instead of the VSAM method, the CPU datapoint does not reflect the execution time spent by the external database server.

If a transaction executes only a few times (small number of Executions) and the CPU datapoint indicates it executes for a longer than expected period of time, the transaction is either a candidate for optimization or the victim of a program logic error. In either case, the application program code requires attention.

CPU System

Description

The CPU (System) datapoint shows the accumulated amount of CPU time the transaction has spent executing operating system calls.

Values

The value is displayed as the total number of seconds.

CPU User

Description

The CPU (User) datapoint shows the accumulated amount of CPU time the transaction has spent executing user (application) code.

Values

The value is displayed as the total number of seconds.

Disabled

Description

The Disabled datapoint indicates whether the transaction is currently disabled.

Values

Value	Description
`true`	Indicates that the transaction is disabled
`false`	Indicates that the transaction is enabled

Interpretation

A disabled transaction cannot be initiated at a Sun MTP region.

Dump Suppressed

Description

The Dump Suppressed datapoint indicates whether the transaction currently has user dumps suppressed.

Values

Value	Description
`true`	Indicates that user dumps are suppressed
`false`	Indicates that user dumps are enabled

Interpretation

When user dumps are suppressed, Sun MTP does not generate a dump file as a result of a transaction abort.

Executions

Description

The Executions datapoint shows the number of times a transaction has executed since the region started.

Values

An integer value initialized to zero at region startup and incremented each time the transaction executes.

Interpretation

The Executions value is the total number of executions of a transaction, whether the transaction was requested internally or externally. Specifically, Executions is the sum of executions due to the following request types: an external request from an end user, internal requests from another program, or an internal request due to a trigger condition for a transient data queue.

This datapoint helps identify high use transactions. The transactions that execute the most often should be targets for optimization to improve overall system performance.

ISC AP Total Requests In

Description

This datapoint represents the number of inbound asynchronous processing (AP) requests for a given transaction received from remote regions.

Values

An integer value initialized to zero at region startup and incremented each time the transaction executes.

Interpretation

There are two different ways that a remote region can initiate an AP request into the local system. First, the remote application program can issue a `START` command with a transaction ID that is defined as remote in the remote region's transaction table. Second, the remote application can specify the `SYSID` as a parameter to the `START` command.

ISC AP Total Requests Out

Description

This datapoint shows the number of asynchronous processing (AP) requests for a particular transaction at a remote region.

Values

An integer value initialized to zero at region startup and incremented each time the transaction executes.

Interpretation

There are two different ways to initiate an AP request to a remote region. First, the application program can issue a `START` command with a transaction ID that is defined as remote in the transaction table. Second, the application can specify the `SYSID` as a parameter to the `START` command.

Program

Description

The Program datapoint contains the name of a system executable that is associated with the transaction name.

Values

The Program datapoint can have a maximum of eight characters.

Interpretation

Every transaction must be associated with the application program that is to be executed on its behalf. Each transaction ID and its associated program is defined in the transaction table.

In order for a region to execute an application program, three conditions must be true. First, the transaction ID and program must be defined in the transaction table. Sun MTP requires this definition to map a request for a particular transaction ID to an actual program. Next, the program must be registered in the program table. Sun MTP refers to this definition to identify the type of file associated with the program. Finally, the file must actually exist.

Remote TranID

Description

The Remote TranID datapoint refers to a transaction that is located on a remote system.

Values

The value of the Remote TranID datapoint is a one- to four-character name representing a transaction ID defined on the remote system.

Interpretation

When a remote transaction ID is specified for a local transaction ID, it indicates that the transaction is to be executed on a remote system and the name of the transaction is specified by the Remote TranID value.

There are two cases that can occur when a remote transaction ID is specified in the transaction table of the local system:

- If a terminal user enters a transaction that is associated with a remote transaction ID (by specifying a value for a remote transaction ID in the transaction table), it is considered transaction routing, that is, the transaction is routed to the remote system for execution.
- If an application program issues a `EXEC CICS START` command and the transaction ID specified is associated with a remote transaction ID in the transaction table, it is considered as an asynchronous transaction request (ISC type AP). This means that the remote transaction will be issued immediately and the local transaction will continue to execute (it will not wait for the remote transaction to complete).

STARTs

Description

The STARTs datapoint displays the number of times a transaction was initiated internally, either by another transaction or application program, or from a transient data queue trigger condition.

Values

The datapoint is an integer value initialized to zero at region startup and incremented as application usage initiates a transaction ID.

Interpretation

This datapoint is similar to the Executions datapoint except that this datapoint does not include the number of times a transaction was requested by an end user directly from a terminal.

Subtract the value in this datapoint from the Executions datapoint for a profile of how often end users at a terminal request a particular transaction.

Screen Size

Description

The Screen Size datapoint shows the terminal screen size allowed for execution of the transaction.

Values

The Screen Size datapoint has one of the following values:

Value	Description
`DEF`	Indicates that the transaction uses the default screen size (24 lines and 80 columns)
`ALT`	Indicates that the transaction uses an alternate screen size

Interpretation

If the Screen Size datapoint is `ALT`, Sun MTP allows this transaction to run on screen sizes of 27 lines and 132 columns (Model 5 3270 Terminal) or 43 lines and 80 columns (Model 4 3270 Terminal). The Screen Size values indicate the screen size that the specified transaction can support.

The Screen Size datapoint can be used to determine if a particular terminal is trying to execute a transaction that it cannot support due to its screen size limitations.

TX Class

Description

The TX Class datapoint contains the name of the transaction class assigned to the transaction.

Values

The TX Class datapoint contains the name of a transaction class configured for the region.

Interpretation

If transaction classes are configured for the region, requests for this transaction will be scheduled by the associated transaction class.

Security Keys

Description

The Security Keys datapoint shows the security level for a particular transaction. An end user must have this security level in order to run the transaction.

Values

A number from 1 to 64. Each number represents a different security level.

Interpretation

Sun MTP compares a transaction's Security Keys value against a user's security key to determine if the user can execute the transaction or not. The administrator may assign a security level to a specific user by defining an entry for the user in the user table.

If a user is not registered in this table, the user receives a default security key of 1. This means the user can only execute transactions with a transaction security value of 1. If the user is in the user table (the system user ID matches the user ID defined in the table), the user implicitly receives the defined security key upon login or can change security levels explicitly (to one specified in the user table) with the `CSSN` transaction.

All Sun MTP system transactions are installed with a default security key value of 1. This enables all users to run these transactions. It is the responsibility of the Sun MTP system administrator to change the security level of the system transactions to limit their use. Refer to the *Sun Mainframe Transaction Processing Software Administrator's Guide* for more information about security.

Transaction Classes

The Transaction Classes window displays information about each transaction class defined in the region.

TABLE 10-16 Transaction Classes Window Datapoints

Datapoint	Description
Name	Identifies the transaction class that can be assigned to a specific transaction resource.
Group	Identifies the group to which the transaction class is assigned.
Peak Transaction Rate	Indicates the maximum number of transactions per second for this transaction class.
Peak Transactions Waiting	Indicates the maximum number of transactions forced to wait for an available transaction processor at the transaction class.
Transaction Processors	Indicates the number of transaction processors assigned to the transaction class.
CPU (Kernel)	Indicates the total system processor (CPU) time for all assigned transaction processors (in seconds).
Transaction Processor Restarts	Indicates the total number of transaction processor restarts.
Transaction Rate	Indicates the number of transactions per second for this transaction class in the last refresh window.
Transactions Aborted	Indicates the total number of transactions that have aborted within the transaction class.
Transactions Deadlocked	Indicates the total number of transactions that have encountered a deadlock within the transaction class.
Transactions Executed	Indicates the total number of transactions that have been executed within the transaction class.
Transactions Waiting	Indicates the number of transaction requests that are currently waiting for an available transaction processor.
CPU (User)	Indicates the total user processor (CPU) time for all assigned transaction processors (in seconds).

Users

The Users pane displays user identification and password information about each user configured for the region. It displays the following datapoints:

TABLE 10-17 Users Window Datapoints

Datapoint	Description
User ID	Defines an identifier for this user that can be used to log in to the region by means of the `CESN` transaction.
Accounting Option	Indicates whether accounting is specified for this user. See "Accounting Option" on page 152.
Password Expiration	Indicates the date and time the existing user password expires.
Group	Identifies the group to which this user is assigned.
Invalid Password	Indicates the total number of times an incorrect password was supplied at login. See "Invalid Password" on page 152.
Operator ID	A three-character datapoint that identifies this user. This information is used by the `EXEC CICS ASSIGN OPID` command.
Operator Class	A three-character datapoint that classifies this user. See "Operator Class" on page 152.
Operator Name	Defines an identifier for this user that can be used to log in to the region via the `CSSN` transaction.
Resource Security Key	Defines the security level for this user for resource management tasks. This information is used by the `EXEC CICS ASSIGN OPERKEYS` command.
Security Key	Defines the security level for this user. See "Security Key" on page 153.
Suspended	Indicates whether the user password is currently suspended. Values are `true` and `false`.

Accounting Option

Description

The Accounting Option datapoint indicates whether accounting is specified for this user.

Values

Value	Description
`Y`	Indicates that accounting is on for this user
`N`	Indicates that accounting is off for this user
`D`	Indicates that accounting is controlled by the user accounting setting in the Monitoring Control Table (MCT)

Interpretation

The value of `Y` for the Accounting Option datapoint does not necessarily mean that the region generates accounting records for a particular user ID. The same condition applies when the Accounting Option datapoint is `D`. Accounting must be specified at both the user and global levels in order for the region to generate accounting records for a user.

Invalid Password

Description

The Invalid Password datapoint indicates the total number of times an incorrect password was supplied when the user tried to log in to the region.

Values

The count is reset to zero following a successful login. The value, therefore, indicates the number of unsuccessful attempts since the last unsuccessful attempt.

Operator Class

Description

The Operator Class datapoint is a three-character datapoint that classifies this user. This information is used by the `EXEC CICS ASSIGN OPCLASS` command.

Values

An Operator Class value of `SYS` enables a user to perform system administration tasks.

Security Key

Description

The Security Key datapoint indicates the security level for this user. This information is used by the `EXEC CICS ASSIGN OPSECURITY` command.

Values

A hexadecimal bit mask that is used to check the user's authority when requesting a transaction.

CHAPTER 11

Advanced Tab

The Advanced tab contains information that is used primarily by customer support personnel. The drop-down list enables you to view the following information types:

Asynchronous Starts

The Asynchronous Starts window displays statistics about the region's start queue. It displays the following datapoints:

TABLE 11-1 Asynchronous Starts Window Datapoints

Datapoint	Description
CICS Request Type	Identifies the request type for this CICS-type message. See "CICS Request Type" on page 159.
Expires	Indicates the time when this request expires.
Facility	Identifies the facility that originated this request (`TASK`, `DEST`). See "Facility" on page 156.
Recoverable	Indicates if the transaction is recoverable. See "Recoverable" on page 157.
Remote Terminal ID	Remote terminal ID assigned to this request. See "Remote Terminal ID" on page 158.
Remote Transaction ID	Remote transaction ID assigned to this request. See "Remote Transaction ID" on page 158.

TABLE 11-1 Asynchronous Starts Window Datapoints *(Continued)*

Datapoint	Description
Remote System ID	System ID assigned to this request. See "Remote System ID" on page 161.
Request ID	Request ID assigned to this request. See "Request ID" on page 159.
Requested	Timestamp when this entry was placed in the start queue. Not implemented in this release.
Start Code	Indicates the transaction start code (`S`, `QD`, `SD`). See "Start Code" on page 160.
Status	Displays the current status of an entry on the start queue. See "Status" on page 160.
Submitting Task	Indicates the exec interface block (EIB) task number that submitted this request. See "Submitting Task" on page 161.
Terminal ID	Terminal ID assigned to this request. See "Terminal ID" on page 162.
Transaction ID	Requested transaction ID. See "Transaction ID" on page 162.
TS Queue Item	Indicates the item number of the temporary storage queue. See "TS Queue Item" on page 157.
TS Queue ID	Identifies the queue assigned to this request. See "TS Queue ID" on page 157.
Type	Indicates the type of entry on the start queue. The only valid value is `CICS`, which represents a CICS statement.

Facility

Description

The Facility datapoint shows the origin of the asynchronously started transaction.

Values

The value of the datapoint is based on the facility type. Facility can be either the transaction ID of the CICS request or the name of the transient data (TD) queue that triggered the request.

Interpretation

Use the Start Code datapoint to determine if the started transaction was requested by another transaction or triggered from a TD queue. When the Start Code datapoint is `S` or `SD`, it indicates that the CICS request came from another transaction. When the Start Code datapoint is `QD`, the request was triggered from a TD queue.

TS Queue Item

Description

The TS Queue Item datapoint displays the number of the item within a temporary storage (TS) queue that was passed with a `START` request.

Values

The value of the datapoint is an integer number that identifies a specific record in a TS queue.

Interpretation

Sun MTP passes data specified with the `FROM` parameter to the `START` statement to the requested transaction by means of the TS Queue Item in a particular TS queue. The TS queue can be specified by the user using the `QUEUE` parameter to the `START` command, or it is generated by Sun MTP. The item number of the data within the TS queue accompanies the entry so that Sun MTP can retrieve the appropriate data.

TS Queue ID

Description

The TS Queue ID datapoint contains the name of a TS queue specified as the `QUEUE` parameter to the `EXEC CICS START` command.

Values

The value of the datapoint is a one- to eight-character queue name.

Interpretation

A particular item in the TS queue holds the data specified by the `FROM` parameter to the `START` command. The TS Queue Item datapoint provides a means to pass data along with the `START` request.

Recoverable

Description

The value of the Recoverable datapoint indicates if the asynchronously started transaction is recoverable or not.

Values

The values of the datapoint can be either `true` or `false`.

Interpretation

If the Recoverable value is `false`, Sun MTP submits the transaction immediately, even for a rollback. However, if the Recoverable value is `true`, Sun MTP does not submit the transaction when there is a rollback. In this sense, the transaction is protected.

Remote Terminal ID

Description

The Remote Terminal ID datapoint contains the value specified by the `RTERMID` parameter of the `EXEC CICS START` command. It represents the terminal name against which the transaction will be run on the remote system.

Values

The value of the datapoint is a one- to four-character name.

Interpretation

The remote terminal name is passed to the asynchronously started transaction, which receives the name by issuing a `RETRIEVE` command with the `RTERMID` option.

Remote Transaction ID

Description

The Remote Transaction ID datapoint represents the value specified by the `RTRANSID` parameter in the `EXEC CICS START` command. It is the remote transaction identifier to be used on the remote system when routing this transaction entry to the remote system.

Values

The value of the datapoint is a one- to four-character name.

Interpretation

The transaction that issues the `START` statement specifies the remote transaction ID. The issued transaction receives the remote transaction ID by issuing a `RETRIEVE` command with the `RTRANSID` option.

The `RTRANSID` parameter contains any data the starting transaction wants to pass to the asynchronously started transaction.

CICS Request Type

Description

The CICS Request Type datapoint shows the type of CICS command request of an entry on the start queue.

Values

The value of the CICS Request Type is:

`START`: Start request (`EXEC CICS START`)

Interpretation

If there is a value in the CICS Request Type datapoint, it indicates that the entry on the start queue originated from a CICS application program. The CICS Request Type datapoint contains a value only when the Type datapoint contains the value `CICS`.

Request ID

Description

The Request ID datapoint displays the value of the `REQID` parameter specified in the `EXEC CICS START` command.

Values

The value of the datapoint can be a one- to eight-character name.

Interpretation

The value of the Request ID datapoint uniquely identifies the associated `START` statement. This value can be used by a `CANCEL` command to cancel the transaction before it executes.

If the Request ID value is specified for the start queue entry, it indicates that the transaction can be cancelled prior to being executed.

Start Code

Description

The Start Code datapoint shows the start code of a START request.

Values

Value	Description
S	START without data
QD	Transient data trigger
SD	START with data

Interpretation

The Start Code datapoint indicates how a request was started (by means of a START command or a transient data trigger) and whether additional data was passed along with the request. When a transaction issues a START request (identified by Task#), the requestor can use the FROM parameter to the START command to specify data to pass along with the request. The transaction requested by the START statement obtains this data by means of the RETREIVE command.

Status

Description

The Status datapoint displays the current status of an entry on the start queue.

Values

Value	Description
0x01	Not yet time to start
0x02	A protected start waiting for requestor to finish
0x04	Terminal is not available
0x08	Not enough background tasks to start this transaction
0x00	Not enough transaction servers available
0x20	More entries on the process queue than there are transaction servers
0x40	ATI bit not set on a terminal
0x80	LU6.2 send or receive failed, task will be rescheduled
81	LU6.2 client did not open a socket with the LU6.2 server
82	The connection is out of service

Value	Description
83	Sun MTP cannot obtain any more shared memory
84	LU6.2 session allocation failed
85	Connection is not defined in the terminal table system entries
Blank	Unknown reason

Interpretation

The Status value of the entry changes based on Sun MTP region resources. If a resource is not available to process the entry, the appropriate value is set for the entry. When the resource is available, the status is reset. Values marked with an asterisk (*) are not resolvable, and will not relinquish their start queue slots.

If a transaction remains in the start queue for a long time, the Status datapoint indicates exactly why the transaction is waiting. As long as the Status datapoint is blank, there is nothing blocking the transaction's eventual execution.

Remote System ID

Description

The Remote System ID datapoint contains the value specified by the `SYSID` parameter in the `EXEC CICS START` command. This is the name of the remote system where the transaction is to run.

Values

The Remote System ID value can be any valid one- to four-character remote system name specified in the system entries portion of the terminal table.

Interpretation

The Remote System ID value can be used to determine the number of jobs that are currently waiting in the start queue that are to be executed on another system.

Submitting Task

Description

The Submitting Task datapoint displays the exec interface block (EIB) task number of the transaction that submitted a transaction to the start queue.

Values

The value of the Submitting Task datapoint is an integer number for a specific instance of a transaction.

Terminal ID

Description

The Terminal ID datapoint contains the value specified by the `TERMID` parameter in the `EXEC CICS START` command. It is the terminal identifier against which the transaction is to be started.

Values

The value of the Terminal ID datapoint is any valid four-character name. If the `START` command is to execute locally, the Terminal ID value must match an entry in the terminal table.

Interpretation

If the Terminal ID datapoint is omitted, the task will run without a terminal, which means that the application program cannot perform I/O to a terminal.

Transaction ID

Description

The Transaction ID datapoint displays the value specified by the `TRANSID` parameter in an `EXEC CICS START` command. The Transaction ID is the transaction identifier.

Values

The value of the Transaction ID datapoint can be any valid four-character transaction identifier specified in the region's transaction table.

Interpretation

The Transaction ID datapoint represents the transaction that was initiated by the end user at the terminal, by a another transaction by means of the `RETURN` statement, or by a transient data queue trigger.

The Transaction ID values on this window give you a good idea of the types of transactions that are waiting to be executed.

Environment Variables

The Environment Variables pane displays each environment variable set for the region and its value.

System Gates

The System Gates panel displays information related to Sun MTP gating primitives. This information is useful for investigating system throughput. The following datapoints are displayed:

TABLE 11-2 System Gates Window Datapoints

Datapoint	Description
Name	Identifies the gate. See "Name" on page 164.
Total Wait Time	Indicates the total time, in seconds, spent attempting to obtain ownership of the gate. This includes any time waiting for the gate due to an existing owner.
Delta Waits	Indicates the delta value of gate waits. See "Delta Waits" on page 165.
Maximum Wait Time	Indicates the maximum time, in seconds, that a requestor was forced to wait for a gate.
Max Lock Time	Indicates the maximum time, in seconds, that an owner held the gate.
Maximum Waiting	Indicates the maximum number of requestors forced to wait for this gate.
Lock Time	Indicates the total amount of time, in seconds, that all owners held this gate.
Owner	Process identifier (PID) of the current owner of the gate.
Total Locks	Indicates the number of times this gate has been owned since the region was started. This value is initialized to zero and region startup and is incremented each time the gate is locked.
Total Waits	Indicates the total number of times a process waited for this gate. See "Total Waits" on page 166
Waiting	Indicates the current number of requestors waiting for this gate.

Name

Description

The Name datapoint shows the name of a system gate used by a region. The server processes of a region employ these gates to control access to shared resources.

Values

The following table describes each gate:

TABLE 11-3 System Gates

Gate Name	Description
KXSEMAL	Allocation gate. Controls access during allocation of blocks of VSAM records.
KXSEMBCA	Browse gate. Controls access to browse control areas in shared memory.
KXSEMBUF(0-f)	VSAM buffer pool gates. The quantity of these gates depends on the number of buffers specified for VSAM datasets in the VCT. The percentage of data buffers versus index buffers depends on the `unikixmain -I` option used. The default percentage is a 50/50 split. Each gate controls access to a subset of the VSAM buffers. `KXSEMBUF0` through `KXSEMBUF7` are always used for the data buffers. `KXSEMBUF8` through `KXSEMBUFf` are always used for index buffers. `KXSEMBUF0` and `KXSEMBUF8` are always valid.
KXSEMENQ	Enqueue gate. Controls access to `CICS ENQ` and `DEQ` memory structures.
KXSEMEVC	Event gate. Controls access to internal Sun MTP counters, such as the transaction number counter that assigns a number to a transaction.
KXSEMFCB	File control gate. Controls access to file control blocks.
KXSEMFLR	Reserved.
KXSEMPPT	Processing program table gate. Controls access to the program table in shared memory.
KXSEMRB	Recovery full gate. Controls access to recovery resources during a recovery file full condition.
KXSEMRCV	Recovery gate. Controls access to recovery resources during before journal processing and dynamic transaction backout.
KXSEMSHR	Shared memory gate. Controls access to shared memory during allocation and deallocation of blocks of memory.
KXSEMSTA	Statistics gate. Controls access during generation of accounting records.

TABLE 11-3 System Gates *(Continued)*

Gate Name	Description
KXSEMSTQ	Start queue gate. Controls access to the start queue. The start queue contains the data retrieved with the CICS RETRIEVE command.
KXSEMTRC	Trace gate. Controls access to the internal Sun MTP trace table.
KXSEMTS	Temporary storage gate. Controls access to the temporary storage queue manipulated by the CICS WRITEQ TS, READQ TS, and DELETEQ TS commands. Both main and auxiliary temporary storage use this gate.
KXSEMUQ	Update gate. Controls access to the update queue during a CICS WRITE, REWRITE, DELETE, or SYNCPOINT. The update queue in Sun MTP contains one entry for each record updated but not yet syncpointed.

Interpretation

A gate is a mechanism that allows multiple processes to gain access to a resource in a mutually exclusive manner. A Sun MTP region's server processes may use all of the gates described in TABLE 11-3. The only user-configurable gates are the buffer and index gates, whose quantity is determined by the number of buffers specified in the VCT. Monitor the usage of the KXSEMBUF0-f gates to adjust the number of buffers for optimum performance. When the numbers of waits on the buffer gates keep climbing, increase the number of buffers to spread out the gate access to the data and indexes in the buffers.

Delta Waits

Description

The Delta Waits datapoint shows the number of times server processes have waited on a particular gate during the last polling interval.

Values

The Delta Waits datapoint is an integer value that is calculated every polling interval.

Interpretation

This datapoint should be monitored when the system seems to be running slowly. A large number of Delta Waits for an extended period of time can indicate a bottleneck in the system due to a particular gate.

Total Waits

Description

The Total Waits datapoint indicates the total number of times a requestor was forced to wait for the gate.

Values

The datapoint value is a number that is initialized to zero at region startup. Sun MTP increments the number of Total Waits for a particular gate every time a requestor has to wait to obtain the gate.

Interpretation

The Total Waits datapoint indicates how much contention a gate has. Two gates may have large numbers of locks, but one may have more waits than the other.

System Queues

The System Queues window displays details about the Sun MTP interprocess communications (IPC) message queues.

Note – The operating system maintains these queues, not the Sun MTP region.

The following datapoints are displayed on this window:

TABLE 11-4 System Queues Window Datapoints

Datapoint	Description
Name	Identifies the queue. See "Name" on page 167.
Maximum Size	Indicates the maximum queue size in bytes. See "Maximum Size" on page 168.
Current Size	Indicates the total number of bytes currently in use by this queue. See "Current Size" on page 169.
Current Messages	Indicates the number of messages currently in this queue. See "Current Messages" on page 169.
Last Receive Process ID	Process ID of process that issued the last `msgrcv` operation.
Last Receive Time	Timestamp of last `msgrcv` operation. See "Last Receive Time" on page 170
Last Send Process ID	Process ID of process that issued the last `msgsnd` operation.

TABLE 11-4 System Queues Window Datapoints *(Continued)*

Datapoint	Description
Last Send Time	Timestamp of last `msgsnd` operation.
Peak Size	Indicates the maximum number of bytes used at one time. See "Peak Size" on page 170.
Peak Messages	Indicates the maximum number of messages queued at one time. See "Peak Messages" on page 171.
Total Bytes	Indicates the total number of bytes processed by this queue.
Total Messages	Indicates the total number of messages processed by this queue.

Name

Description

The Name datapoint shows the name of a system queue used by Sun MTP. A queue is an operating system resource enabling interprocess communication. Sun MTP creates a number of queues that are used to pass messages and requests between the various components and server processes that make up a region.

Values

The possible values for the Name datapoint are: `KXPRTQ`, `KXPROCQ`, `KXRECOVQ`, `KXSTRTQ`, `KXTERMQ`, `KXCLASSQ`*nnn*, and `KXTRANQ`*nnn*.

Interpretation

The following table describes each type of queue:

TABLE 11-5 System Queues

Queue	Description
`KXPRTQ`	The print server uses the print queue to process the Sun MTP log files: `unikixmain.log`, `unikixmain.err`, and `unikixmain.dbg`.
`KXPROCQ`	The process queue is used to initiate all new transactions. Normally, messages are placed on this queue by the TN3270 server, `unikixtnemux`. Messages can also be placed on this queue by the start queue server (`EXEC CICS START`), a transient data queue trigger, or a batch job. Any transaction server may remove messages from this queue.
`KXRECOVQ`	The recovery queue is used to process recovery requests, including saving before images or rolling back transactions.
`KXSTRTQ`	The start queue is used to process interval control requests. These requests come from the following `EXEC CICS` commands: `START`, `POST`, `WAIT`, `DELAY`, or `CANCEL`.

TABLE 11-5 System Queues *(Continued)*

Queue	Description
KXTERMQ	The terminal queue is used by the EPI facility to pass a message from a transaction server to an EPI process.
KXTRANQ*nnn*	The number of transaction queues is the same as the number of transaction servers. Unlike the process queue, where messages are processed by any transaction server, each transaction server has its own transaction queue to process server-specific messages, such as requests to read/write data and recovery requests. Each transaction queue has a number (*nnn*) corresponding to a transaction server.
KXCLASSQ*nnn*	One queue per transaction class. The `unikixsched` process removes messages from the process queue and schedules them to the appropriate class queue. Each class queue has a number (*nnn*), which is the queue ID.

The most important queues relating to system performance are KXPROCQ, KXCLASSQ, and KXRECOVQ.

Maximum Size

Description

The Maximum Size datapoint represents the maximum number of bytes a system queue can hold at any one time. A queue is an operating system resource enabling interprocess communication. This value is part of your operating system configuration (kernel parameter).

Values

This numeric value is obtained directly from the operating system, not from the region.

Interpretation

If, for some reason, the capacity of a queue is reached, the process trying to put a message or request on the queue must wait until another process takes a message or request off the queue. This condition can cause region throughput to slow down. You can detect this condition by comparing the Peak Size datapoint to the corresponding Maximum Size datapoint. If the Peak Size datapoint reaches the allowed size of the queue, the queue size must be increased.

Refer to your operating system's system administrator's guide for information about increasing the maximum size of a system queue.

Monitor the Maximum Size datapoint, especially during the initial deployment of a Sun MTP region, to help determine an appropriate queue size. Resizing system queues not only requires shutting down the region, but the operating system as well, affecting all activity on the machine.

Current Size

Description

The Current Size datapoint represents the number of bytes currently on a particular system queue. A queue is an operating system resource enabling interprocess communication.

Values

The value of the Current Size datapoint is a numeric quantity obtained directly from the operating system.

Interpretation

The value of the Current Size datapoint indicates how heavily a queue is used. The value is the total number of bytes used by messages on a queue. The number of messages appears in the Current Messages datapoint.

The Current Size value is a snapshot in time: at each refresh interval, the administration tool queries the operating system for the size of the queue.

When the value of the Peak Size datapoint approaches the maximum size for the process queue, the transaction class queue, or any of the transaction server queues, more transaction servers can be configured to alleviate server overload for the region. Examine the Performance screens to see how the transaction servers are being utilized. If transactions are always being run in the transaction servers during peak production hours and messages are always waiting to be processed on the process queue or transaction class queues, increase the number of transaction servers.

Current Messages

Description

The Current Messages datapoint represents the number of messages or requests currently on a particular system queue. A queue is an operating system resource enabling interprocess communication.

Values

The value of the Current Messages datapoint is a numeric quantity obtained directly from the operating system.

Interpretation

A message on a queue consists of a certain number of bytes. Each message on a particular queue is the same size, but each queue can have its own message size. To determine the message size for a queue, divide the number of Current Size by the number of Current Messages. Dividing the Maximum Size by the size of a message tells you the effective maximum number of messages allowed on a particular queue.

The value of Current Messages on the process queue and transaction class queues tells how many requests are waiting to be processed. If this number remains high as response time diminishes, consider configuring more transaction servers to accommodate the message processing.

The value of Current Messages on the recovery queue indicates the number of requests (before image, writes, or rollbacks) that are waiting to be processed.

Last Receive Time

Description

The Last Receive Time datapoint shows the last time a process took an entry from a system queue. A queue is an operating system resource enabling interprocess communication.

Values

The Last Receive Time datapoint has the following format: *hh:mm:ss*. This value is obtained directly from the operating system.

Interpretation

This statistic can be used to determine if processes are taking messages or requests from the queue in a timely manner.

Consider a situation where the Current Size value reaches the allowed size. This indicates that the queue membership has reached its capacity in terms of size. A normal course of action could be to increase the size of the queue or adjust the number of processes that read the queue. However, the Last Receive Time datapoint might reveal that processes are not taking messages off the queue for some reason, such as system problems, aborted processes, looping processes, and so on. Make sure to evaluate all possible causes before changing your configuration.

Peak Size

Description

The Peak Size datapoint represents the maximum number of bytes that have been recorded on a particular system queue. A queue is an operating system resource enabling interprocess communication.

Values

The value of the Peak Size datapoint is a numeric quantity obtained by comparing the current value to the value of Current Size and setting Peak Size to the higher of the two.

Interpretation

This value indicates the queue usage (along with the Peak Messages datapoint) at peak load in production processing. Note that the actual peak usage of a queue might occur between polling intervals and could escape the administration tool's inquiries to the operating system.

If the value of the Peak Size datapoint approaches the value of the Maximum Size datapoint, consider increasing the size of the queue.

Peak Messages

Description

The Peak Messages datapoint indicates the maximum number of messages that have been recorded on a particular system queue. A queue is an operating system resource enabling interprocess communication.

Values

The value of the Peak Messages datapoint is a numeric quantity obtained by comparing the current value to the value of Current Messages and setting Peak Messages to be the maximum of the two.

Interpretation

This value indicates the queue usage (along with the Peak Size datapoint) at peak loads in production processing. The actual peak usage of a queue might occur between polling intervals and may escape the administration tool's inquiries to the operating system.

Most systems have a kernel parameter that specifies the maximum number of messages that can be on a queue at any given time. This parameter is similar to the Maximum Size datapoint of a queue (which is measured in bytes). Consult your operating system's system administrator's guide to find out where to look for this value. If the Peak Messages datapoint approaches this value, it must be increased to obtain better performance from the region.

The size of the messages put on the queue can also affect the maximum number of messages allowed on a queue, because of a queue's maximum size. For example, if a queue has an allowed size of 100 bytes and a maximum of 10 messages, and the messages for this queue are 50 bytes each, the queue has an effective limit of two messages.

Index

E

F

G

H

I

J

K

T

U

V

W