IBM Tivoli Decision Support for z/OS
Version 1.8.2

Distributed Systems Performance Feature Guide and Reference

SH19-4018-13

IBM Tivoli Decision Support for z/OS
Version 1.8.2

Distributed Systems Performance Feature Guide and Reference

SH19-4018-13

Before using this information and the product it supports, read the information in "Notices" on page 217.

Thirteenth Edition (March 2015)

This edition applies to version 1, release 8, modification level 2 of Tivoli Decision Support for z/OS (program number 5698-B06) and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Figures

Preface

The *Distributed Systems Performance Feature Guide and Reference*, SH19-4018 manual describes how to use IBM® Tivoli® Decision Support for z/OS® (hereafter referred to as Tivoli Decision Support for z/OS) to collect and report performance data generated by operating systems and applications running on a workstation. This book:

- Describes performance issues and how they affect the level of services you can offer users
- Guides you through the component-selection, installation, and implementation processes
- Explores performance characteristics shown in Tivoli Decision Support for z/OS reports so that you can analyze the characteristics of your system

Note: The term UNIX used in this book covers AIX, HP-UX, and Solaris operating systems. When AIX is used, this refers only to the IBM operating system (and not to HP-UX or Solaris).

The terms MVS™, OS/390®, and z/OS are used interchangeably throughout this book:

Who should read this book

The *Distributed Systems Performance Feature Guide and Reference*, SH19-4018 is intended for:

- Anyone who analyzes or monitors distributed systems performance.
- Anyone responsible for establishing or meeting service-level objectives for distributed systems.
- Tivoli Decision Support for z/OS administrators (primarily as a guide to feature installation and as a reference to table and report definitions)
- Users with various backgrounds who are interested in analyzing and improving distributed systems performance.

What this book contains

This book explains how to collect distributed systems performance data, and to create and display Tivoli Decision Support for z/OS reports to both monitor and understand performance on these platforms.

It is broadly divided into two sections.

Parts 1 and 2 detail the newer distributed component which replaces the deprecated UNIX and Linux components.

Because the distributed component uses more modern tooling for development, it's more easily portable to new platfoms, and the ability to configure external tools for data collection allows easier adaptation to both newer platforms and current platforms where the environment changes.

The Windows component has also been copied here as that will remain after the deprecated UNIX and Linux components are retired.

Parts 3 and 4 cover the UNIX and Linux components which are still supported, but deprecated. Newer installations should choose the distributed component. It also covers the Windows component which has been duplicated in parts 1 and 2.

This book contains the following parts:

- Use Part 1 for developing and implementing your performance and service-level strategy if you wish to use the newer distributed component, or the Windows component:
  - Chapter 1, "Introducing the Distributed Systems Performance feature," on page 3 explains basic concepts of systems management and offers suggestions on establishing performance objectives and service-level agreements. It also describes the Distributed Systems Performance feature role in the Tivoli Decision Support for z/OS environment.
  - Chapter 2, "Installing and configuring the Distributed component," on page 7 contains a task-oriented description of how to plan for and set up the distributed component so that useful reports and decision-support information are available immediately.
  - Chapter 3, "Installing and configuring the Windows component," on page 33 contains a task-oriented description of how to plan for and set up the Windows components so that useful reports and decision-support information are available immediately.
- Use Part 2 for a detailed description of the distributed component:
  - Chapter 4, "Data flow and Tivoli Decision Support for z/OS objects," on page 45 describes the flow of data from log record to report, showing Tivoli Decision Support for z/OS log and record definitions, tables, and reports.
  - Chapter 5, "Data tables," on page 47 describes the supplied data tables and updates, including columns and expressions. It contains similar descriptions of lookup tables and control tables. This chapter also describes the data tables used by the Distributed Systems Performance component.
  - Chapter 6, "Reports," on page 53 contains examples of the Distributed Systems Performance component reports and shows where the data comes from.
  - Chapter 7, "Log record definitions," on page 73 describes the Distributed Systems Performance component record definitions and lists the record types you can select with each of them.
- Use Part 3 for developing and implementing your performance and service-level strategy if you wish to use the deprecated UNIX and Linux components or the Windows component:
  - Chapter 8, "Introducing the Distributed Systems Performance feature," on page 81 explains basic concepts of systems management and offers suggestions on establishing performance objectives and service-level agreements. It also describes the Distributed Systems Performance feature role in the Tivoli Decision Support for z/OS environment.
  - Chapter 9, "Installing and configuring the UNIX and Linux Performance components," on page 85 contains a task-oriented description of how to plan for and set up the UNIX and Linux components so that useful reports and decision-support information are available immediately.
  - Chapter 10, "Installing and configuring the Windows component," on page 117 contains a task-oriented description of how to plan for and set up the Windows components so that useful reports and decision-support information are available immediately.

- Use Part 4 for a detailed description of the UNIX, Linux and Windows components:
  - Chapter 11, "Data flow and Tivoli Decision Support for z/OS objects," on page 129 describes the flow of data from log record to report, showing Tivoli Decision Support for z/OS log and record definitions, tables, and reports.
  - Chapter 12, "Data, lookup, and control tables," on page 133 describes the supplied data tables and updates, including columns and expressions. It contains similar descriptions of lookup tables and control tables. This chapter also describes the data tables used by the Distributed Systems Performance component.
  - Chapter 14, "Reports," on page 151 contains examples of the Distributed Systems Performance component reports and shows where the data comes from.
  - Chapter 15, "Log record definitions," on page 201 describes the Distributed Systems Performance component record definitions and lists the record types you can select with each of them.
- Use Part 5 as a reference for obtaining support for IBM software products.

A glossary and index follow the appendixes.

Publications

This section lists publications in the Tivoli Decision Support for z/OS library and any other related documents. It also describes how to access Tivoli publications online, how to order Tivoli publications, and how to submit comments on Tivoli publications.

Tivoli Decision Support for z/OS library

The following documents are available in the Tivoli Decision Support for z/OS library:

- *Administration Guide and Reference*, SH19-6816.

  Provides information about initializing the Tivoli Decision Support for z/OS database and customizing and administering Tivoli Decision Support for z/OS.
- *AS/400 System Performance Feature Guide and Reference*, SH19-4019.

  Provides information for administrators and users about collecting and reporting performance data generated by AS/400 systems.
- *CICS Performance Feature Guide and Reference*, SH19-6820.

  Provides information for administrators and users about collecting and reporting performance data generated by Customer Information and Control System (CICS®).
- *Distributed Systems Performance Feature Guide and Reference*, SH19-4018.

  Provides information for administrators and users about collecting and reporting performance data generated by operating systems and applications running on a workstation.
- *Guide to Reporting*, SH19-6842.

  Provides information for users who display existing reports, for users who create and modify reports, and for administrators who control reporting dialog default functions and capabilities.
- *IMS Performance Feature Guide and Reference*, SH19-6825.

  Provides information for administrators and users about collecting and reporting performance data generated by Information Management System (IMS).

- *Language Guide and Reference*, SH19-6817.

  Provides information for administrators, performance analysts, and programmers who are responsible for maintaining system log data and reports.
- *Messages and Problem Determination*, SH19-6902.

  Provides information to help operators and system programmers understand, interpret, and respond to Tivoli Decision Support for z/OS messages and codes.
- *Network Performance Feature Installation and Administration*, SH19-6901.

  Provides information for network analysts or programmers who are responsible for setting up the network reporting environment.
- *Network Performance Feature Reference*, SH19-6822.

  Provides reference information for network analysts or programmers who use the Network Performance feature.
- *Network Performance Feature Reports*, SH19-6821.

  Provides information for network analysts or programmers who use the Network Performance feature reports.
- *Resource Accounting for z/OS*, SH19-4495.

  Provides information for users who want to use Tivoli Decision Support for z/OS to collect and report performance data generated by Resource Accounting for z/OS.
- *Resource Accounting*, SH19-6818.

  Provides information for performance analysts and system programmers who are responsible for meeting the service-level objectives established in your organization.
- *System Performance Feature Guide*, SH19-6819.

  Provides information for administrators and users with a variety of backgrounds who want to use Tivoli Decision Support for z/OS to analyze z/OS, z/VM®, zLinux, and their subsystems, performance data.
- *System Performance Feature Reference Volume I*, SH19-4494.

  Provides information for administrators and users with a variety of backgrounds who want to use Tivoli Decision Support for z/OS to analyze z/OS, z/VM, zLinux, and their subsystems, performance data.
- *System Performance Feature Reference Volume II*, SC23-7966.

  Provides information about the functions and features of the Usage and Accounting Collector.
- *IBM Online Library z/OS Software Products Collection Kit*, SK3T-4270

  CD containing all z/OS documentation.

Accessing terminology online

The IBM Terminology Web site consolidates the terminology from IBM product libraries in one convenient location. You can access the Terminology Web site at the following Web address:

http://www.ibm.com/ibm/terminology

Accessing publications online

IBM posts publications for this and all other Tivoli products, as they become available and whenever they are updated, to the IBM Knowledge Center Web site. Access the Tivoli Decision Support for z/OS web page in the Knowledge Center by going to the following address: https://www-01.ibm.com/support/knowledgecenter/SSH53X/welcome

Accessibility

Accessibility features help users with a physical disability, such as restricted mobility or limited vision, to use software products successfully. With this product, you can use assistive technologies to hear and navigate the interface. You can also use the keyboard instead of the mouse to operate all features of the graphical user interface.

For additional information, see the Accessibility Appendix in the *Administration Guide and Reference*.

Tivoli technical training

For Tivoli technical training information, refer to the following IBM Tivoli Education Web site:

http://www.ibm.com/software/tivoli/education/

Support information

If you have a problem with your IBM software, you want to resolve it quickly. IBM provides the following ways for you to obtain the support you need:

- Searching knowledge bases: You can search across a large collection of known problems and workarounds, Technotes, and other information.
- Obtaining fixes: You can locate the latest fixes that are already available for your product.
- Contacting IBM Software Support: If you still cannot solve your problem, and you need to work with someone from IBM, you can use a variety of ways to contact IBM Software Support.

For more information about these three ways of resolving problems, see "Support information," on page 213.

Conventions used in this book

This guide uses several conventions for special terms and actions, operating system-dependent commands and paths, and margin graphics.

The following terms are used interchangeably throughout this book:
- MVS, OS/390, and z/OS.
- VM and z/VM.

Except for editorial changes, updates to this edition are marked with a vertical bar to the left of the change.

Typeface conventions

This guide uses the following typeface conventions:

Bold

- Lowercase commands and mixed case commands that are otherwise difficult to distinguish from surrounding text
- Interface controls (check boxes, push buttons, radio buttons, spin buttons, fields, folders, icons, list boxes, items inside list boxes,

multicolumn lists, containers, menu choices, menu names, tabs, property sheets), labels (such as **Tip**, and **Operating system considerations**)

- Column headings in a table
- Keywords and parameters in text

Italic

- Citations (titles of books, diskettes, and CDs)
- Words defined in text
- Emphasis of words (words as words)
- Letters as letters
- New terms in text (except in a definition list)
- Variables and values you must provide

Monospace

- Examples and code examples
- File names, programming keywords, and other elements that are difficult to distinguish from surrounding text
- Message text and prompts addressed to the user
- Text that the user must type
- Values for arguments or command options

Part 1. Distributed Systems Performance Feature Guide

Chapter 1. Introducing the Distributed Systems Performance feature

IBM Tivoli Decision Support for z/OS (hereafter referred to as Tivoli Decision Support for z/OS) is a reporting system that collects performance data logged by computer systems, summarizes the data, and presents it in a variety of forms for use in systems management. Tivoli Decision Support for z/OS consists of a base product and several optional features.

The Distributed Systems Performance feature is a solution for environments where z/OS is used, and where distributed operating systems (such as AIX, HP-UX, Solaris, Linux, and Windows) are installed on nodes within your network. The performance information from the various nodes is transferred to a central z/OS site, where reporting and analysis are performed.

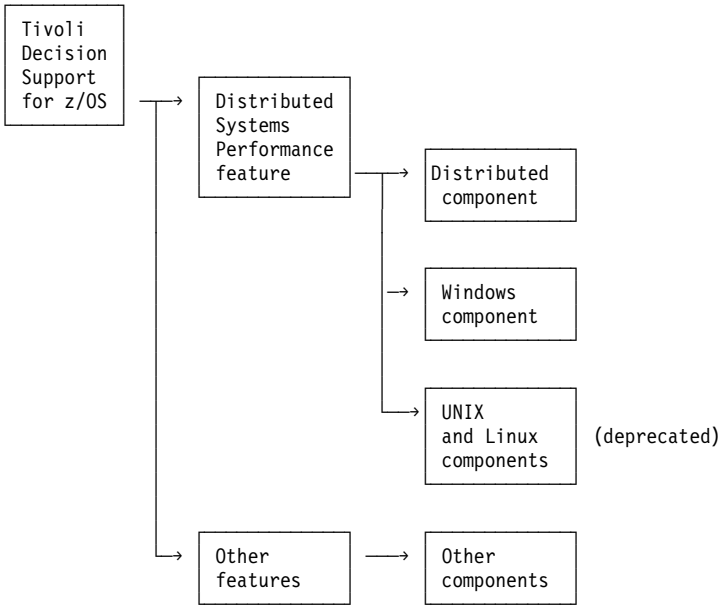
The Distributed component is the newer style component for collecting information from most distributed platforms.

The Windows component is the equivalent component for collecting information from Windows platforms.

The heritage UNIX and Linux components are also used for collecting information from distributed platforms. However, all future development effort will go towards the Distributed and Windows components.

Parts 1 and 2 of this document refer only to the Distributed and Windows components. For details on the heritage UNIX and Linux components, please see parts 3 and 4.

To see how the product, features, and components relate to one another, see the diagram below.



The advantages of the Distributed component over the older UNIX and Linux components are as follows:

- The code that runs on the distributed nodes uses more modern, cross-platform development tools so that it can be more easily ported to new environments. This was first pioneered in the Windows component and has now been implemented for the other distributed platforms.
- The use of these development tools also removes a significant impediment to fast turn-around of bug fixes caused by the requirement to test changes across all platforms. Because the tools present a common environment, the code can be common across all platforms, current and future.
- All low-level operating system data collection tasks are now externalized in the configuration file, allowing for easy changes in the event the tasks change their output formats. This capability was earlier put into the Linux and zLinux agents and means that changes to the underlying operating system should not need a new code release, only a configuration change.
- The ability for the distributed nodes to generate both the newer versioned log record format (for the Distributed component) and the older style (for the UNIX and Linux components). This will aid in transition.
- The ability to run as a non-root user on the distributed nodes, greatly reducing security risks.
- Versioning of log record definitions so that distributed nodes can be upgraded gradually.
- Removal of wastage in the log record formats, leading to reduced storage requirements.
- Can collect both standard records as defined in this document.

This topic describes how information is gathered and collected using the Distributed Systems Performance feature.

Gathering and collecting performance data

At the nodes where Distributed and Windows component agents have been installed and from which data is to be used, the basic information used by the Distributed Systems Performance feature is *gathered* by the agents into log files containing various performance and other data. The source data used by the Distributed Systems Performance feature to create the log files and a description of the log files, is given here:

Source data
Description of created log files

Distributed data
Log files containing processor use, I/O rate, and paging space data. These are continuously updated by programs supplied with the Distributed Systems Performance feature.

Windows data
Log files containing various statistics collected from Windows systems. These are continuously updated by programs supplied with the Distributed Systems Performance feature.

SMF104 data
SMF104 data

The information contained in the log files must be transmitted to z/OS for use as data sets in the Tivoli Decision Support for z/OS COLLECT procedure. The most

convenient way is to use a product such as TCP/IP. The `ftp` function of TCP/IP can be used to automate logon and file transfer.

After the transmission of log files is completed, the information contained in log data sets at the z/OS host is *collected* into Tivoli Decision Support for z/OS tables.

The collected data is combined with more data (called *environment data*) and is finally presented in reports.

The process of entering and maintaining environment data is called *administration*. Tivoli Decision Support for z/OS provides an administration dialog for maintaining resource information. Refer to the Administration Guide and Referencemanual for information on how to use the administration dialog.

Figure 1 illustrates how data is organized for presentation in Tivoli Decision Support for z/OS reports.

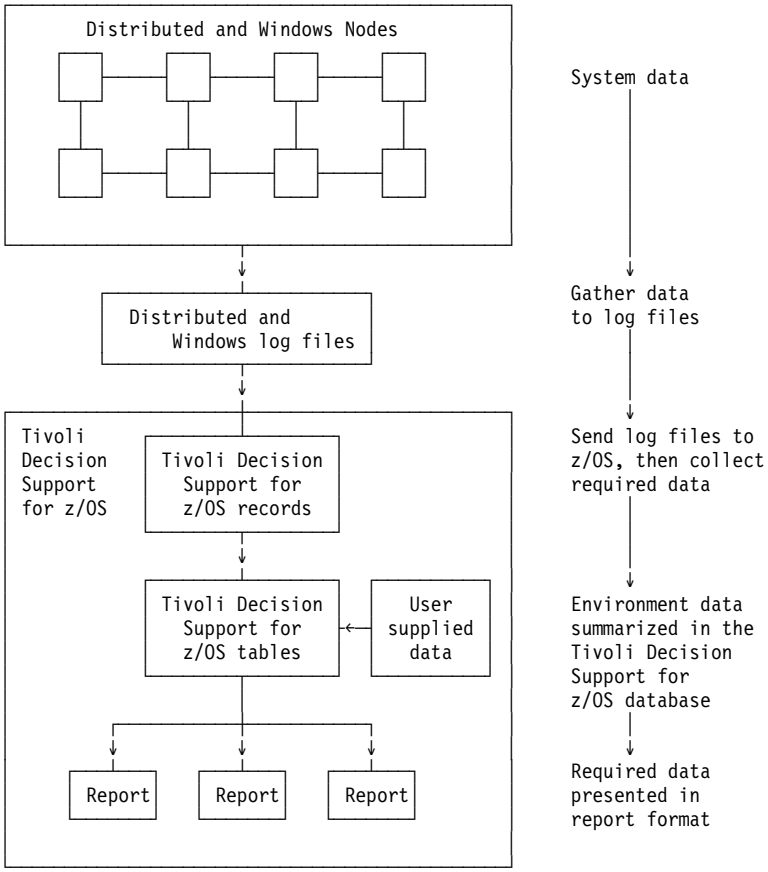


Figure 1. Organizing and presenting system performance data

Report Groups

The reports produced by the Distributed Systems Performance feature are grouped in the following report groups:

Distributed
Distributed component reports

Windows
Windows component reports

Report Groups

The reports cover a wide range of needs in a data processing center, and reporting can be done online or in batch. They are accessible from the Tivoli Decision Support for z/OS reporting dialog.

Finally, the key to successful implementation of Tivoli Decision Support for z/OS is knowing:

- The information and resources on which you want to report and how to perform customization to select them
- The way you want to organize, set objectives for, and process the data (used later to define the environment)

Chapter 2. Installing and configuring the Distributed component

This supplements the procedure in the Administration Guide and Reference for installing a component, with information specific to the Distributed component.

The chaptertopic describes how to plan, install, test and monitor the Distributed component.

Planning the implementation process

About this task

Before installing the Distributed component, you should follow these steps to plan the implementation process:

Procedure

1. Describe user tasks. Then determine what data the Distributed component must gather to help users accomplish those tasks.
2. Determine the administration tasks you must perform for the Distributed component, and make any decisions required by these tasks. These tasks help you customize Tivoli Decision Support for z/OS and the Distributed component to work efficiently and effectively with your computer system.

Results

If this is your first exercise in implementation planning, *follow all these steps* to ensure that the Distributed component's implementation is consistent. If you are reading this chaptertopic in preparation for modifying your system, you might not need to perform all of these tasks.

Use the planning process to prepare for these main customization tasks:

- Customizing the distributed nodes to generate the data required by the Distributed component.
- Defining environment data, which is all the information (in addition to the input data) that the Distributed component needs to create reports. Environment data controls the data collection process and provides more information in the reports.

Figure 2 illustrates the process for implementing the Distributed component.

Figure 2. Implementation process for the Distributed component

Installing and configuring your distributed nodes

About this task

These sections give you step-by-step information on how to install and configure Tivoli Decision Support for z/OS on your distributed nodes. This section gives a general overview of the process, and the sections following this give more detailed descriptions for each target platform.

The general process consists of the following steps:

Procedure

- Check that the requirements are met, in terms of hardware, software, and other items.
- Prepare for, and install, the software on the distributed node.
- Configure the distributed node for running the software.
- Start the software on the distributed node.
- Configure the distributed node for automatically running the software at startup time.

Results

Instructions for each platform can be found in the following sections:

- "Installing and configuring your SLES nodes," for SuSe Linux Enterprise Server 11.
- "Installing and configuring your RHEL nodes" on page 12, for Red Hat Enterprise Linux 6.
- "Installing and configuring your HP-UX nodes" on page 16, for HP-UX 11i v3.
- "Installing and configuring your AIX nodes" on page 21, for IBM AIX 7.1
- "Installing and configuring your Solaris nodes" on page 25, for Oracle Solaris 11.1

Note: Although the environments above are the only current officially supported platforms at the time of publication, the Distributed component has been built with portability in mind. The work required to get it running on other platforms will be minor compared to the heritage UNIX and Linux components.

Installing and configuring your SLES nodes

About this task

This section gives you practical step-by-step information on how to install and configure Tivoli Decision Support for z/OS on your SLES nodes.

Step 1: Checking requirements (SLES)

About this task

Check the following requirements:

- "Disk space requirements"
- "Software requirements" on page 9

Disk space requirements

You need the following approximate disk space on each distributed node:

Table 1. Disk space requirements on each distributed node

Installation task	Disk space (approximate)
Load code in the file system	1MB
Create work files during the gathering process	5MB

Note: The 5MB of work files consists of:

Table 2. Detals of gathering process disk space requirements

Record type	Space required
File systems (DISKFSA)	48KB (based on log record size of 100 bytes, gathering once per hour, and 20 disks)
Paging spaces (PERFPSA)	12KB (based on log record size of 100 bytes, gathering once per hour, and 5 spaces)
Processor and memory information (PERFVMA)	28KB (based on log record size of 200 bytes, gathering every 10 minutes)
Disk I/O information (PERFIOA)	576KB (based on log record size of 200 bytes, gathering every 10 minutes, and 20 disks)
Auxiliary amounts	4MB (temporary space needs, and allowance for several days of failed transmissions)

Software requirements

The Distributed component requires:

- Oracle SuSE Linux Enterprise Server 11.
- **IMPORTANT:** SLES does not install the `iostat` program by default. You should first check to see if it is installed by attempting to run `iostat` from a shell. If it cannot be found, it should be installed by the root user with the command `yast2 --install sysstat`.

Step 2: Preparing and installing Tivoli Decision Support for z/OS code on the distributed node (SLES)

About this task

The distributed part of the Distributed component is contained in the following SMP target library member:`DRL`*xxx*`.SDRLWS(DRLDIST)`

Download and install the`tar.gz` file using these steps

Procedure

1. As the `root` user, create a new user `drluser`, according to the processes detailed by your nominated operating system. SLES has the `useradd` command for this purpose. To do this:
   a. Login as the `root` user.
   b. Execute the command `useradd -m drluser`.
   c. Execute the command `passwd drluser`.
   d. Enter your password, then enter password again for confirmation.
2. Once the user is created (and the password is set), login as that user.
3. Enter the following commands from your home directory (`/home/drluser`) .
   - `mkdir drl`
   - `cd drl`
4. Start an FTP session to your z/OS host, and receive the file from the host where you installed Tivoli Decision Support for z/OS. In the following example, the highlighted entries show the commands you must enter.

   zosbox
   The IP address/name of the z/OS system where Tivoli Decision Support for z/OS was installed and where the `tar.gz` file is located.

tsoid The TSO user ID that has read access to the `tar.gz` file.

secret The password for the TSO user ID.

drlbase
The high-level qualifier where Tivoli Decision Support for z/OS is installed.

The following transcript shows the transfer in action.

```
drluser(/home/drluser/drl): ftp zosbox
Connected to zosbox.corp.com.
220-FTPSERVE at ZOSBOX.CORP.COM, 11:05:24 on 02/24/12
220 Connection will close if idle for more than 5 minutes.
Name (zosbox:drluser): tsoid
331 Send password please.
Password: secret
230 TSOID is logged on.
ftp> binary
200 Representation type is IMAGE.
ftp> get 'drlbase.SDRLWS(DRLDIST)' drldist.tar.gz
200 Port request OK.
125 Sending data set DRL181.SDRLWS(DRLDIST) FIXrecfm 128
12345 bytes received
ftp> quit
221 Quit command received. Goodbye.
drluser(/home/drluser/drl):
```

5. On the node to which the tar file has been received, enter the following commands to unpack the `drldist.tar.gz` file and create the files in the directories:

   ```
   tar zxvf drldist.tar.gz
   ```

   This will create and populate the `bin`, `var` and `etc` subdirectories:
   - `bin` holds all the executable files for the distributed code.
   - `var` holds all the data files created by the distributed code for transmission to the host, along with all temporary files.
   - `etc` holds all the configuration files for the distributed code.

Step 3: Configuring the distributed node (SLES)

About this task

Configure your node by following these steps:

Procedure

1. Set up control files to control environment and permissions. Do this by editing the relevant start files for the `drluser` user. Under SLES, this is the profile in your home directory, `/home/drluser/.profile`. Add the following commands to the end of this file:

   ```
   export PATH=$PATH:$HOME/drl/bin:/sbin
   export DRLCFG=$HOME/drl/etc/drl.cfg
   umask 077
   chmod 755 $HOME/drl $HOME/drl/bin $HOME/drl/bin/*
   chmod 700 $HOME/drl/doc $HOME/drl/doc/*
   chmod 700 $HOME/drl/etc $HOME/drl/etc/*
   chmod 700 $HOME/drl/var $HOME/drl/var/*
   ```

   Then log out and log back in as `drluser`
2. Configure the `/home/drluser/drl/etc/drl.cfg` configuration file. As a minimum, you need to change the following parameters:

Table 3. Parameters that need to be configured

Parameter	Description
General:SysName	The system name for identification purposes in Tivoli Decision Support for z/OS. This should be unique across all distributed nodes.
Xfer:Host	The host name or IP address of the host to transmit the data to.
Xfer:User	The TSO user ID at the host.
Xfer:Password	The password for that TSO user ID at the host.
Perf:IoCmd	The prefabricated command indicator for collecting disk I/O statistics. Should be `@0000` for SLES.
Perf:VmCmd	The prefabricated command indicator for collecting memory statistics. Should be `@0000` for SLES.
Perf:UserCmd	The prefabricated command indicator for collecting user statistics. Should be `@0000` for SLES.
Perf:ProcCmd	The prefabricated command indicator for collecting process statistics. Should be `@0000` for SLES.
Disk:FsCmd	The prefabricated command indicator for collecting file system statistics. Should be `@0000` for SLES.
Page:PsCmd	The prefabricated command indicator for collecting paging space statistics. Should be `@0000` for SLES.

3. You may also want to examine the other parameters in the configuration file to see how they affect various aspects. Each parameter section is documented within the file itself. Specifically, you may need to change `Xfer:DsName` to change where the data will be stored on the host. To do this, change the first bit from `TDS` to the high-level qualifier you want for your data.

Step 4: Starting the distributed node (SLES)

About this task

Start your node by following these steps:

Procedure

1. While logged in as the `drluser` user, startup the gathering processes with the command `drlctrl start`.
2. Ensure that the following processes are running (assuming you have not disabled them by changing `General:RunProcs` parameter in the configuration file):

   `drlmntr`
   The monitor process responsible for keeping everything running.

   `drlperf`
   The process responsible for collecting disk I/O, memory usage, user, and process information.

   `drldisk`
   The process responsible for collecting disk usage information.

   `drlpage`
   The process responsible for collecting paging space information.

drlxfer
The process responsible for transferring all information to the host.

drlclnp
The process responsible for cleaning up old files.

3. Check the `nhp` and `log` files in the `var` directory for errors or warnings.
4. Ensure that the `out` files are being created in the `var` directory (for the `drlperf`, `drldisk` and `drlpage` processes). It may take some time for all of these to appear; 10 minutes in the default configuration.
5. Ensure that the `drlxfer` process successfully transfers files to the host. In the default configuration this will not happen until early the following day, and can be checked by examining the log files after the attempt.
6. Ensure that the `drlclnp` process successfully clears files from the node. This can be tested by creating a `log.drlperf-1900-01-01` in the `var` directory, then executing the command `drlctrl kill clnp`, and ensuring that the file is deleted.

Step 5: Configuring auto start on the distributed node (SLES)

About this task

Configure the distributed node to automatically start the gathering software on startup by following these steps. These have to be executed by the `root` user since they are changing system processes.

Procedure

1. Login as `root` and go to the system startup directory `cd /etc/init.d`
2. Create the file `drluser` using an editor. The contents of this file should be
   ```
   #!/bin/sh
   su - drluser "/home/drluser/drl/bin/rc.drluser"
   ```
3. Change the permissions of the startup file with `chmod 755 drluser`.
4. Edit the `/etc/inittab` file and add the line `drl:5:wait:/etc/init.d/drluser`.
5. If possible, reboot the node to ensure the processes start up.

Installing and configuring your RHEL nodes

About this task

This section gives you step-by-step information on how to install and configure Tivoli Decision Support for z/OS on the RHEL nodes.

Step 1: Checking requirements (RHEL)

About this task

Check the following requirements:

- "Disk space requirements" on page 13
- "Software requirements" on page 13

Disk space requirements
About this task

You need the following approximate disk space on each distributed node:

Table 4. Disk space requirements on each distributed node

Installation task	Disk space (approximate)
Loading code in the file system	1MB
Creating work files during the gathering process	5MB

Note: The 5MB of work files consists of:

Table 5. Details of gathering process disk space requirements

Record type	Space required
File systems (DISKFSA)	48KB (based on log record size of 100 bytes, gathering once per hour, and 20 disks)
Paging spaces (PERFPSA)	12KB (based on log record size of 100 bytes, gathering once per hour, and 5 spaces)
Processor and memory information (PERFVMA)	28KB (based on log record size of 200 bytes, gathering every 10 minutes)
Disk I/O information (PERFIOA)	576KB (based on log record size of 200 bytes, gathering every 10 minutes, and 20 disks)
Auxiliary amounts	4MB (temporary space needs, and allowance for several days of failed transmissions)

Software requirements
About this task

The Distributed component requires:

- Red Hat Linux Enterprise Server 6.

Step 2: Preparing and installing Tivoli Decision Support for z/OS code on the distributed node (RHEL)
About this task

The distributed part of the Distributed component is contained in the following SMP target library member:`DRL`*`xxx`*`.SDRLWS(DRLDIST)`

Download and install the `tar.gz` file using these steps:

Procedure

1. As the `root` user, create a new user `drluser`, according to the processes detailed by your nominated operating system. RHEL has the `useradd` command for this purpose. To do this:
   a. Login as the `root` user.
   b. Execute the command `useradd -m drluser`.
   c. Execute the command `passwd drluser`.
   d. Enter the desired password, then enter password again for confirmation.
2. Once the user is created (and the password is set), login as that user.

3. Enter the following commands from your home directory (/home/drluser):
   - mkdir drl
   - cd drl
4. Start an FTP session to your z/OS host, and receive the file from the host where you installed Tivoli Decision Support for z/OS. In the following example, the highlighted entries show the commands you must enter.

   zosbox
   The IP address/name of the z/OS system where Tivoli Decision Support for z/OS was installed and where the tar.gz file is located.

   tsoid The TSO user ID that has read access to the tar.gz file.

   secret The password for the TSO user ID.

   drlbase
   The high-level qualifier where Tivoli Decision Support for z/OS is installed.

   The following transcript shows the transfer in action.

```
drluser(/home/drluser/drl): ftp zosbox
Connected to zosbox.corp.com.
220-FTPSERVE at ZOSBOX.CORP.COM, 11:05:24 on 02/24/12
220 Connection will close if idle for more than 5 minutes.
Name (zosbox:drluser): tsoid
331 Send password please.
Password: secret
230 TSOID is logged on.
ftp> binary
200 Representation type is IMAGE.
ftp> get 'drlbase.SDRLWS(DRLDIST)' drldist.tar.gz
200 Port request OK.
125 Sending data set DRL181.SDRLWS(DRLDIST) FIXrecfm 128
12345 bytes received
ftp> quit
221 Quit command received. Goodbye.
drluser(/home/drluser/drl):
```

5. On the node to which the tar file has been received, enter the following commands to unpack the drldist.tar.gz file and create the files in the directories:

   ```
   tar zxvf drldist.tar.gz
   ```

   This will create and populate the bin, var and etc subdirectories:
   - bin holds all the executable files for the distributed code.
   - var holds all the data files created by the distributed code for transmission to the host, along with all temporary files.
   - etc holds all the configuration files for the distributed code.

Step 3: Configuring the distributed node (RHEL)

About this task

Configure your node by following these steps:

Procedure

1. Set up control files to control environment and permissions. This is performed by editing the relevant startup files for the drluser user. Under RHEL, this is the profile in the user's home directory, /home/drluser/.bash_profile. Add the following commands to the end of this file:

```
export PATH=$PATH:$HOME/drl/bin:/sbin
export DRLCFG=$HOME/drl/etc/drl.cfg
umask 077
chmod 755 $HOME/drl $HOME/drl/bin $HOME/drl/bin/*
chmod 700 $HOME/drl/doc $HOME/drl/doc/*
chmod 700 $HOME/drl/etc $HOME/drl/etc/*
chmod 700 $HOME/drl/var $HOME/drl/var/*
```

Then log out and log back in as `drluser`.

2. Configure the `/home/drluser/drl/etc/drl.cfg` configuration file. As a minimum you need to change the following parameters:

Table 6. Parameters that need to be configured

Parameter	Description
General:SysName	The system name for identification purposes in Tivoli Decision Support for z/OS. This should be unique across all distributed nodes.
Xfer:Host	The host name or IP address of the host to which the data will transmitted.
Xfer:User	The TSO user ID at the host.
Xfer:Password	The password for that TSO user ID at the host.
Perf:IoCmd	The prefabricated command indicator for collecting disk I/O statistics. Should be `@0000` for RHEL.
Perf:VmCmd	The prefabricated command indicator for collecting memory statistics. Should be `@0000` for RHEL.
Perf:UserCmd	The prefabricated command indicator for collecting user statistics. Should be `@0000` for RHEL.
Perf:ProcCmd	The prefabricated command indicator for collecting process statistics. Should be `@0000` for RHEL.
Disk:FsCmd	The prefabricated command indicator for collecting file system statistics. Should be `@0000` for RHEL.
Page:PsCmd	The prefabricated command indicator for collecting paging space statistics. Should be `@0000` for RHEL.

3. You may also want to examine the other parameters in the configuration file to see how they affect various aspects. Each parameter section is documented within the file itself. Specifically, you may need to change `Xfer:DsName` in order to change where the data will be stored on the host. To do this, change the first bit from `TDS` to whatever high-level qualifier you want for your data.

Step 4: Starting the distributed node (RHEL)

About this task

Start your node by following these steps:

Procedure

1. While logged in as the `drluser` user, startup the gathering processes with the command `drlctrl start`.
2. Ensure that the following processes are running (assuming you have not disabled them by changing `General:RunProcs` parameter in the configuration file):

   `drlmntr`
   The monitor process responsible for keeping everything running.

drlperf
The process responsible for collecting disk I/O, memory usage, user and process information.

drldisk
The process responsible for collecting disk usage information.

drlpage
The process responsible for collecting paging space information.

drlxfer
The process responsible for transferring all information to the host.

drlclnp
The process responsible for cleaning up old files.

3. Check the `nhp` and `log` files in the `var` directory for errors or warnings.
4. Ensure that the `out` files are being created in the `var` directory (for the `drlperf`, `drldisk` and `drlpage` processes). It may take some time for all of these to appear; 10 minutes in the default configuration.
5. Ensure that the `drlxfer` process successfully transfers files to the host. In the default configuration this will not happen until early the following day, and can be checked by examining the log files after the attempt.
6. Ensure that the `drlclnp` process successfully clears files from the node. This can be tested by creating a `log.drlperf-1900-01-01` in the `var` directory, then executing the command `drlctrl kill clnp`, and ensuring that the file is deleted.

Step 5: Configuring auto-start on the distributed node (RHEL)

About this task

Configure the distributed node to automatically start the gathering software on startup by following these steps. These have to be executed by the `root` user as you are changing system processes.

Procedure

1. Login as `root` and go to the system startup directory `cd /etc/init.d`.
2. Create the file `drluser` using an editor. The contents of this file should be:
```
#!/bin/sh
su - drluser "/home/drluser/drl/bin/rc.drluser"
```
3. Change the permissions of the startup file with `chmod 755 drluser`.
4. Create a symbolic link for the startup file:
```
cd /etc/rc5.d
ln -s /etc/init.d/drluser S99drluser
```
5. If possible, reboot the node to ensure the processes start up.

Installing and configuring your HP-UX nodes

About this task

This section gives you step-by-step information on how to install and configure Tivoli Decision Support for z/OS on your HP-UX nodes.

Step 1: Checking requirements (HP-UX)

About this task

Check the following requirements:

- "Disk space requirements"
- "Software requirements"

Disk space requirements

About this task

You need the following approximate disk space on each distributed node:

Table 7. Disk space requirements on each distributed node

Installation task	Disk space (approximate)
Loading code in the file system	1MB
Creating work files during the gathering process	5MB

Note: The 5MB of work files consists of:

Table 8. Detals of gathering process disk space requirements

Record type	Space required
File systems (DISKFSA)	48KB (based on log record size of 100 bytes, gathering once per hour, and 20 disks)
Paging spaces (PERFPSA)	12KB (based on log record size of 100 bytes, gathering once per hour, and 5 spaces)
Processor and memory information (PERFVMA)	28KB (based on log record size of 200 bytes, gathering every10 minutes)
Disk I/O information (PERFIOA)	576KB (based on log record size of 200 bytes, gathering every 10 minutes, and 20 disks)
Auxiliary amounts	4MB (temporary space needs, and allowance for several days of failed transmissions)

Software requirements

About this task

The Distributed component requires:

- HP-UX 11i v3.

Step 2: Preparing and installing Tivoli Decision Support for z/OS code on the distributed node (HP-UX)

About this task

The distributed part of the Distributed component is contained in the following SMP target library member:`DRL`*`xxx`*`.SDRLWS(DRLDIST)`

Download and install this the `tar.gz` file using these steps:

Procedure

1. As the root user, create a new user drluser, according to the processes detailed by your nominated operating system. HP-UX has the useradd command for this purpose. To do this
   a. Login as the root user.
   b. Execute the command useradd -m drluser.
   c. Execute the command passwd drluser.
   d. Enter the desired password, then enter password again for confirmation.
2. Once the user is created (and the password is set), login as that user.
3. Enter the following commands from your home directory (/home/drluser) .
   - mkdir drl
   - cd drl
4. Start an FTP session to your z/OS host, and receive the file from the host where you installed Tivoli Decision Support for z/OS. In the following example, the highlighted entries show the commands you must enter.

   zosbox
   The IP address/name of the z/OS system where Tivoli Decision Support for z/OS was installed and where the tar.gz file is located.

   tsoid The TSO user ID that has read access to the tar.gz file.

   secret The password for the TSO user ID.

   drlbase
   The high-level qualifier where Tivoli Decision Support for z/OS is installed.

   The following transcript shows the transfer in action.

```
drluser(/home/drluser/drl): ftp zosbox
Connected to zosbox.corp.com.
220-FTPSERVE at ZOSBOX.CORP.COM, 11:05:24 on 02/24/12
220 Connection will close if idle for more than 5 minutes.
Name (zosbox:drluser): tsoid
331 Send password please.
Password: secret
230 TSOID is logged on.
ftp> binary
200 Representation type is IMAGE.
ftp> get 'drlbase.SDRLWS(DRLDIST)' drldist.tar.gz
200 Port request OK.
125 Sending data set DRL181.SDRLWS(DRLDIST) FIXrecfm 128
12345 bytes received
ftp> quit
221 Quit command received. Goodbye.
drluser(/home/drluser/drl):
```

5. On the node to which the tar file has been received, enter the following commands to unpack the drldist.tar.gz file and create the files in the directories:

```
gunzip drldist.tar.gz
tar xvf drldist.tar
```

   This will create and populate the bin, var and etc subdirectories:
   - bin holds all the executable files for the distributed code.
   - var holds all the data files created by the distributed code for transmission to the host, along with all temporary files.
   - etc holds all the configuration files for the distributed code.

Step 3: Configuring the distributed node (HP-UX)

About this task

Configure your node by following these steps:

Procedure

1. Set up control files to control environment and permissions. This is performed by editing the relevant startup files for the `drluser` user. Under HP-UX, this is the profile in the user's home directory, `/home/drluser/.profile`. Add the following commands to the end of this file

   ```
   export PATH=$PATH:$HOME/drl/bin:/usr/sbin
   export DRLCFG=$HOME/drl/etc/drl.cfg
   umask 077
   chmod 755 $HOME/drl $HOME/drl/bin $HOME/drl/bin/*
   chmod 700 $HOME/drl/doc $HOME/drl/doc/*
   chmod 700 $HOME/drl/etc $HOME/drl/etc/*
   chmod 700 $HOME/drl/var $HOME/drl/var/*
   ```

   Then log out and log back in as `drluser`.
2. Configure the `/home/drluser/drl/etc/drl.cfg` configuration file. As a minimum, you need to change the following parameters:

Table 9. Parameters that need to be configured

Parameter	Description
General:SysName	The system name for identification purposes in Tivoli Decision Support for z/OS. This should be unique across all distributed nodes.
Xfer:Host	The host name or IP address of the host to transmit the data to.
Xfer:User	The TSO user ID at the host.
Xfer:Password	The password for that TSO user ID at the host.
Perf:IoCmd	The prefabricated command indicator for collecting disk I/O statistics. Should be `@0001` for HP-UX.
Perf:VmCmd	The prefabricated command indicator for collecting memory statistics. Should be `@0001` for HP-UX.
Perf:UserCmd	The prefabricated command indicator for collecting user statistics. Should be `@0000` for HP-UX.
Perf:ProcCmd	The prefabricated command indicator for collecting process statistics. Should be `@0000` for HP-UX.
Disk:FsCmd	The prefabricated command indicator for collecting file system statistics. Should be `@0000` for HP-UX.
Page:PsCmd	The prefabricated command indicator for collecting paging space statistics. Should be `@0001` for HP-UX.

3. You may also want to examine the other parameters in the configuration file to see how they affect various aspects. Each parameter section is documented within the file itself. Specifically, you may need to change `Xfer:DsName` to change where the data will be stored on the host. Most likely, this will simply be changing the first bit from `TDS` to whatever high level qualifier you want for your data.

Step 4: Starting the distributed node (HP-UX)

About this task

Start your node by following these steps:

Procedure

1. While logged in as the `drluser` user, startup the gathering processes with the command `drlctrl start`.
2. Ensure that the following processes are running (assuming you haven't disabled them by changing `General:RunProcs` parameter in the configuration file):

   drlmntr
   The monitor process responsible for keeping everything running.

   drlperf
   The process responsible for collecting disk I/O, memory usage, user, and process information.

   drldisk
   The process responsible for collecting disk usage information.

   drlpage
   The process responsible for collecting paging space information.

   drlxfer
   The process responsible for transferring all information to the host.

   drlclnp
   The process responsible for cleaning up old files.
3. Check the `nhp` and `log` files in the `var` directory for errors or warnings.
4. Ensure that the `out` files are being created in the `var` directory (for the `drlperf`, `drldisk` and `drlpage` processes). It may take some time for all of these to appear; 10 minutes in the default configuration.
5. Ensure that the `drlxfer` process successfully transfers files to the host. In the default configuration this will not happen until early the following day, and can be checked by examining the log files after the attempt.
6. Ensure that the `drlclnp` process successfully clears files from the node. This can be tested by creating a `log.drlperf-1900-01-01` in the `var` directory, then executing the command `drlctrl kill clnp` and ensuring that the file is deleted.

Step 5: Configuring auto-start on the distributed node (HP-UX)

About this task

Configure the distributed node to automatically start the gathering software on startup by following these steps. These have to be executed by the `root` user since they are changing system processes.

Procedure

1. Login as `root` and go to the system startup directory `cd /sbin/init.d`.
2. Create the file `drluser` using an editor. The contents of this file should be:

```
#!/bin/sh
su - drluser "/home/drluser/drl/bin/rc.drluser"
```

3. Change the permissions of that startup file with `chmod 755 drluser`.
4. Edit the `/etc/inittab` file and add the line `drl:345:wait:/sbin/init.d/drluser`.
5. If possible, reboot the node to ensure the processes start.

Installing and configuring your AIX nodes

About this task

This section gives you practical step-by-step information on how to install and configure Tivoli Decision Support for z/OS on your AIX nodes.

Step 1: Checking requirements (AIX)

About this task

Check the following requirements:

- "Disk space requirements"
- "Software requirements"

Disk space requirements

About this task

You need the following approximate disk space on each distributed node:

Table 10. Disk space requirements on each distributed node

Installation task	Disk space (approximate)
Loading code in the file system	1MB
Creating work files during the gathering process	5MB

Note: The 5MB of work files consists of:

Table 11. Detals of gathering process disk space requirements

Record type	Space required
File systems (DISKFSA)	48KB (based on log record size of 100 bytes, gathering once per hour, and 20 disks)
Paging spaces (PERFPSA)	12KB (based on log record size of 100 bytes, gathering once per hour, and 5 spaces)
Processor and memory information (PERFVMA)	28KB (based on log record size of 200 bytes, gathering every 10 minutes)
Disk I/O information (PERFIOA)	576KB (based on log record size of 200 bytes, gathering every 10 minutes, and 20 disks)
Auxiliary amounts	4MB (temporary space needs, and allowance for several days of failed transmissions)

Software requirements

About this task

The Distributed component requires:

- IBM AIX 7.1.

Step 2: Preparing and installing Tivoli Decision Support for z/OS code on the distributed node (AIX)

About this task

The distributed part of the Distributed component is contained in the following SMP target library member: DRL*xxx*.SDRLWS(DRLDIST)

Download and install this tar.gz file using these steps:

Procedure

1. As the root user, create a new user drluser, according to the processes detailed by your nominated operating system. HP-UX has the useradd command for this purpose. To do this
   a. Login as the root user.
   b. Execute the command useradd -m drluser.
   c. Execute the command passwd drluser.
   d. Enter the desired password, then enter password again for confirmation.
2. Once the user is created (and the password is set), login as that user.
3. Enter the following commands from your home directory (/home/drluser) .
   - mkdir drl
   - cd drl
4. Start an FTP session to your z/OS host, and receive the file from the host where you installed Tivoli Decision Support for z/OS. In the following example, the highlighted entries show the commands you must enter.

   zosbox
   The IP address/name of the z/OS system where Tivoli Decision Support for z/OS was installed and where the tar.gz file is located.

   tsoid The TSO user ID that has read access to the tar.gz file.

   secret The password for the TSO user ID.

   drlbase
   The high-level qualifier where Tivoli Decision Support for z/OS is installed.

   The following transcript shows the transfer in action.

```
drluser(/home/drluser/drl): ftp zosbox
Connected to zosbox.corp.com.
220-FTPSERVE at ZOSBOX.CORP.COM, 11:05:24 on 02/24/12
220 Connection will close if idle for more than 5 minutes.
Name (zosbox:drluser): tsoid
331 Send password please.
Password: secret
230 TSOID is logged on.
ftp> binary
200 Representation type is IMAGE.
ftp> get 'drlbase.SDRLWS(DRLDIST)' drldist.tar.gz
200 Port request OK.
125 Sending data set DRL181.SDRLWS(DRLDIST) FIXrecfm 128
12345 bytes received
ftp> quit
221 Quit command received. Goodbye.
drluser(/home/drluser/drl):
```

5. On the node to which the tar file has been received, enter the following commands to unpack the drldist.tar.gz file and create the files in the directories:

```
gunzip drldist.tar.gz
tar xvf drldist.tar
```

This will create and populate the `bin`, `var` and `etc` subdirectories:

- `bin` holds all the executable files for the distributed code.
- `var` holds all the data files created by the distributed code for transmission to the host, along with all temporary files.
- `etc` holds all the configuration files for the distributed code.

Step 3: Configuring the distributed node (AIX)

About this task

Configure your node by following these steps:

Procedure

1. Set up control files to control environment and permissions. This is performed by editing the relevant startup files for the `drluser` user. Under HP-UX, this is the profile in the user's home directory, `/home/drluser/.profile`. Add the following commands to the end of this file

   ```
   export PATH=$PATH:$HOME/drl/bin:/usr/sbin
   export DRLCFG=$HOME/drl/etc/drl.cfg
   umask 077
   chmod 755 $HOME/drl $HOME/drl/bin $HOME/drl/bin/*
   chmod 700 $HOME/drl/doc $HOME/drl/doc/*
   chmod 700 $HOME/drl/etc $HOME/drl/etc/*
   chmod 700 $HOME/drl/var $HOME/drl/var/*
   ```

   Then log out and log back in as `drluser`.
2. Configure the `/home/drluser/drl/etc/drl.cfg` configuration file. At a bare minimum, you need to change the following parameters:

Table 12. Parameters that need to be configured

Parameter	Description
General:SysName	The system name for identification purposes in Tivoli Decision Support for z/OS. This should be unique across all distributed nodes.
Xfer:Host	The host name or IP address of the host to transmit the data to.
Xfer:User	The TSO user ID at the host.
Xfer:Password	The password for that TSO user ID at the host.
Perf:IoCmd	The prefabricated command indicator for collecting disk I/O statistics. Should be `@0002` for AIX.
Perf:VmCmd	The prefabricated command indicator for collecting memory statistics. Should be `@0002` for AIX.
Perf:UserCmd	The prefabricated command indicator for collecting user statistics. Should be `@0000` for AIX.
Perf:ProcCmd	The prefabricated command indicator for collecting process statistics. Should be `@0000` for AIX.
Disk:FsCmd	The prefabricated command indicator for collecting file system statistics. Should be `@0001` for AIX.
Page:PsCmd	The prefabricated command indicator for collecting paging space statistics. Should be `@0002` for AIX.

3. You may also want to examine the other parameters in the configuration file to see how they affect various aspects. Each parameter section is documented within the file itself. Specifically, you may need to change `Xfer:DsName` to change where the data will be stored on the host. Most likely, this will simply be changing the first bit from `TDS` to whatever high level qualifier you want for your data.

Step 4: Starting the distributed node (AIX)

About this task

Start your node by following these steps:

Procedure

1. While logged in as the `drluser` user, startup the gathering processes with the command `drlctrl start`.
2. Ensure that the following processes are running (assuming you haven't disabled them by changing `General:RunProcs` parameter in the configuration file):

   drlmntr
   The monitor process responsible for keeping everything running.

   drlperf
   The process responsible for collecting disk I/O, memory usage, user, and process information.

   drldisk
   The process responsible for collecting disk usage information.

   drlpage
   The process responsible for collecting paging space information.

   drlxfer
   The process responsible for transferring all information to the host.

   drlclnp
   The process responsible for cleaning up old files.
3. Check the `nhp` and `log` files in the `var` directory for errors or warnings.
4. Ensure that the `out` files are being created in the `var` directory (for the `drlperf`, `drldisk` and `drlpage` processes). It may take some time for all of these to appear; 10 minutes in the default configuration.
5. Ensure that the `drlxfer` process successfully transfers files to the host. In the default configuration this will not happen until early the following day, and can be checked by examining the log files after the attempt.
6. Ensure that the `drlclnp` process successfully clears files from the node. This can be tested by creating a `log.drlperf-1900-01-01` in the `var` directory, then executing the command `drlctrl kill clnp` and ensuring that the file is deleted.

Step 5: Configuring auto-start on the distributed node (AIX)

About this task

Configure the distributed node to automatically start the gathering software on startup by following these steps. These have to be executed by the `root` user since they are changing system processes.

Procedure

1. Login as `root` and go to the system startup directory `cd /etc/rc.d/init.d`.
2. Create the file `drluser` using an editor. The contents of this file should be:

   ```
   #!/usr/bin/sh
   su - drluser "/home/drluser/drl/bin/rc.drluser"
   ```

3. Change the permissions of the startup file with `chmod 755 drluser`.
4. Create a symbolic link for the startup file

   ```
   cd /etc/rc.d/rc2.d
   ln -s /etc/rc.d/init.d/drluser S999drluser
   ```

5. If possible, reboot the node to ensure the processes start.

Installing and configuring your Solaris nodes

About this task

This section gives you step-by-step information on how to install and configure Tivoli Decision Support for z/OS on your Solaris nodes.

Step 1: Check requirements (Solaris)

About this task

Check the following requirements:

- "Disk space requirements"
- "Software requirements" on page 26

Disk space requirements

About this task

You need the following approximate disk space on each distributed node:

Table 13. Disk space requirements on each distributed node

Installation task	Disk space (approximate)
Loading code in the file system	1MB
Creating work files during the gathering process	5MB

Note: The 5MB of work files consists of:

Table 14. Detals of gathering process disk space requirements

Record type	Space required
File systems (DISKFSA)	48KB (based on log record size of 100 bytes, gathering once per hour, and twenty disks)
Paging spaces (PERFPSA)	12KB (based on log record size of 100 bytes, gathering once per hour, and five spaces)
Processor and memory information (PERFVMA)	28KB (based on log record size of 200 bytes, gathering every ten minutes)
Disk I/O information (PERFIOA)	576KB (based on log record size of 200 bytes, gathering every ten minutes, and twenty disks)
Auxiallary amounts	4MB (temporary space needs, and allowance for several days of failed transmissions)

Software requirements
About this task

The Distributed component requires:

- Oracle Solaris 11.1

Step 2: Preparing and installing Tivoli Decision Support for z/OS code on the distributed node (Solaris)
About this task

The distributed part of the Distributed component is contained in the following SMP target library member: `DRL`*xxx*`.SDRLWS(DRLDIST)`

Download and install this `tar.gz` file using these steps:

Procedure

1. As the `root` user, create a new user `drluser`, according to the processes detailed by your nominated operating system. Solaris has the `useradd` command for this purpose. To do this
   a. Login as the `root` user.
   b. Execute the command `useradd -m drluser`.
   c. Execute the command `passwd drluser`.
   d. Enter the desired password, then enter password again for confirmation.
2. Once the user is created (and the password is set), login as that user.
3. Enter the following commands from your home directory (`/export/home/drluser`) .
   - `mkdir drl`
   - `cd drl`
4. Start an FTP session to your z/OS host, and receive the file from the host where you installed Tivoli Decision Support for z/OS. In the following example, the highlighted entries show the commands you must enter.

   zosbox
   The IP address/name of the z/OS system where Tivoli Decision Support for z/OS was installed and where the `tar.gz` file is located.

   tsoid The TSO user ID that has read access to the `tar.gz` file.

   secret The password for the TSO user ID.

   drlbase
   The high-level qualifier where Tivoli Decision Support for z/OS is installed.

   The following transcript shows the transfer in action.

```
drluser(/home/drluser/drl): ftp zosbox
Connected to zosbox.corp.com.
220-FTPSERVE at ZOSBOX.CORP.COM, 11:05:24 on 02/24/12
220 Connection will close if idle for more than 5 minutes.
Name (zosbox:drluser): tsoid
331 Send password please.
Password: secret
230 TSOID is logged on.
ftp> binary
200 Representation type is IMAGE.
ftp> get 'drlbase.SDRLWS(DRLDIST)' drldist.tar.gz
200 Port request OK.
125 Sending data set DRL181.SDRLWS(DRLDIST) FIXrecfm 128
12345 bytes received
ftp> quit
221 Quit command received. Goodbye.
drluser(/home/drluser/drl):
```

5. On the node to which the tar file has been received, enter the following commands to unpack the `drldist.tar.gz` file and create the files in the directories:

   ```
   gunzip drldist.tar.gz
   tar xvf drldist.tar
   ```

   This will create and populate the `bin`, `var` and `etc` subdirectories:
   - `bin` holds all the executable files for the distributed code.
   - `var` holds all the data files created by the distributed code for transmission to the host, along with all temporary files.
   - `etc` holds all the configuration files for the distributed code.

Step 3: Configure the distributed node (Solaris)

About this task

Configure your node by following these steps:

Procedure

1. Set up control files to control environment and permissions. This is performed by editing the relevant startup files for the `drluser` user. Under Solaris, this is the profile in the user's home directory, `/export/home/drluser/.profile`. Add the following commands to the end of this file:

   ```
   export PATH=/usr/xpg4/bin:$PATH:$HOME/drl/bin:/usr/sbin
   export DRLCFG=$HOME/drl/etc/drl.cfg
   umask 077
   chmod 755 $HOME/drl $HOME/drl/bin $HOME/drl/bin/*
   chmod 700 $HOME/drl/doc $HOME/drl/doc/*
   chmod 700 $HOME/drl/etc $HOME/drl/etc/*
   chmod 700 $HOME/drl/var $HOME/drl/var/*
   ```

   Then log out and log back in as `drluser`.
2. Configure the `/export/home/drluser/drl/etc/drl.cfg` configuration file. At a bare minimum, you need to change the following parameters:

Table 15. Parameters that need to be configured

Parameter	Description
General:SysName	The system name for identification purposes in Tivoli Decision Support for z/OS. This should be unique across all distributed nodes.

Table 15. Parameters that need to be configured (continued)

Parameter	Description
Xfer:Host	The host name or IP address of the host to transmit the data to.
Xfer:User	The TSO user ID at the host.
Xfer:Password	The password for that TSO user ID at the host.
Perf:IoCmd	The prefabricated command indicator for collecting disk I/O statistics. Should be `@0003` for Solaris.
Perf:VmCmd	The prefabricated command indicator for collecting memory statistics. Should be `@0003` for Solaris.
Perf:UserCmd	The prefabricated command indicator for collecting user statistics. Should be `@0000` for Solaris.
Perf:ProcCmd	The prefabricated command indicator for collecting process statistics. Should be `@0000` for Solaris.
Disk:FsCmd	The prefabricated command indicator for collecting file system statistics. Should be `@0000` for Solaris.
Page:PsCmd	The prefabricated command indicator for collecting paging space statistics. Should be `@0003` for Solaris.

3. You may also want to examine the other parameters in the configuration file to see how they affect various aspects. Each parameter section is documented within the file itself. Specifically, you may need to change `Xfer:DsName` to change where the data will be stored on the host. Most likely, this will simply be changing the first bit from `TDS` to whatever high level qualifier you want for your data.
4. In particular, ensure that you check the entries for `Path:BinPath` and `Path:VarPath`. The default Solaris install places home directories into `/export/home` rather than just `/home`.

Step 4: Start and check the distributed node (Solaris)

About this task

Start your node by following these steps:

Procedure

1. While logged in as the `drluser` user, start up the gathering processes with the command `drlctrl start`.
2. Ensure that the following processes are running (assuming you haven't disabled them by changing `General:RunProcs` parameter in the configuration file):
   - `drlmntr`, the monitor process responsible for keeping everything running.
   - `drlperf`, the process responsible for collecting disk I/O, memory usage, user and process information.
   - `drldisk`, the process responsible for collecting disk usage information.
   - `drlpage`, the process responsible for collecting paging space information.
   - `drlxfer`, the process responsible for transferring all information to the host.
   - `drlclnp`, the process responsible for cleaning up old files.
3. Check the `nhp` and `log` files in the `var` directory for errors or warnings.

4. Ensure that the `out` files are being created in the `var` directory (for the `drlperf`, `drldisk` and `drlpage` processes). Note that it may take some time for all of these to appear, ten minutes in the default configuration.
5. Ensure that the `drlxfer` process successfully transfers files to the host. In the default configuration, this will not happen until early the following day, and can be checked by examining the log files after the attempt.
6. Ensure that the `drlclnp` process successfully clears files from the node. This can be tested by creating a `log.drlperf-1900-01-01` in the `var` directory, then executing the command `drlctrl kill clnp` and ensuring that the file is deleted.

Step 5: Configure auto-start on the distributed node (Solaris)

About this task

Configure the distributed node to automatically start the gathering software on startup by following these steps. Note that these have to be executed by the `root` user since they are changing system processes.

Procedure

1. Login as `root` and go to the system startup directory `cd /etc/init.d`.
2. Create the file `drluser` using an editor. The contents of this file should be:

   ```
   #!/usr/bin/sh
   su - drluser "/export/home/drluser/drl/bin/rc.drluser"
   ```

3. Change the permissions of that startup file with `chmod 755 drluser`.
4. Create a symbolic link for the startup file:

   ```
   cd /etc/rc3.d
   ln -s /etc/init.d/drluser S999drluser
   ```

5. Modify the `/export/home/drluser/drl/bin/rc.drluser` file to change the `DRLCFG` variable to `/export/home/drluser/drl/etc/drl.cfg`.
6. If possible, reboot the node to ensure the processes start.

Installing Distributed component on the z/OS system

About this task

When you install the Distributed component, Tivoli Decision Support for z/OS will install the required log and record definitions, record procedures, and update definitions to the product system tables. Tivoli Decision Support for z/OS will also install the predefined tables (described in Chapter 5, “Data tables,” on page 47) and reports (described in Chapter 6, “Reports,” on page 53). To install the Distributed component, use the Administration dialog.

Perform steps 1 to 4, as follows.

Procedure

1. From the Tivoli Decision Support for z/OS Administration window (Figure 3 on page 30), select 2, Components and press Enter.

```
  Other  Utilities  Help
--------------------------------------------------------------------------------
                            Administration
Command ===> ____________________________________________________________

Select one of the following.  Then press Enter.


              2_  1.  System
                  2.  Components
                  3.  Logs
                  4.  Tables
                  5.  Reports


 F1=Help      F2=Split     F3=Exit      F9=Swap     F10=Actions  F12=Cancel
```

The Components window is displayed, (as shown in Figure 4).
Figure 3. Tivoli Decision Support for z/OS Administration window

```
  Component  Space  Other  Help
--------------------------------------------------------------------------------
                               Components                  ROW 1 TO 16 OF 80
Command ===> ____________________________________________________________

Select one or more components. Then press Enter to Open component.

/   Components                                          Status     Date
_   z/OS System (MVS) Component                     Installed  2012-09-12
_   z/OS Availability Component
_   z/OS Interval Job/Step Accounting Component
_   z/OS Performance Management (MVSPM) Component
_   z/VM Performance Component
_   CICS Monitoring Component
_   CICS Monitoring Partitioned Component
_   CICS OMEGAMON Monitoring Component
_   CICS Statistics Component
_   CICS Statistics Partitioned Component
_   CICS Tran and UOW Analysis Partitioning Component
_   CICS Transaction and UOW Analysis Component
_   CICS Transaction Gateway Component
_   Data Set Component
/   Distributed Component
_   DB2 Component

 F1=Help      F2=Split     F3=Exit      F5=New       F6=Install   F7=Bkwd
 F8=Fwd       F9=Swap     F10=Actions  F12=Cancel
```

Figure 4. Components window

2. From the Components window, select the components to install (here, the Distributed component) and press F6.
3. The Installation Options window is displayed, (as shown in Figure 5 on page 31).

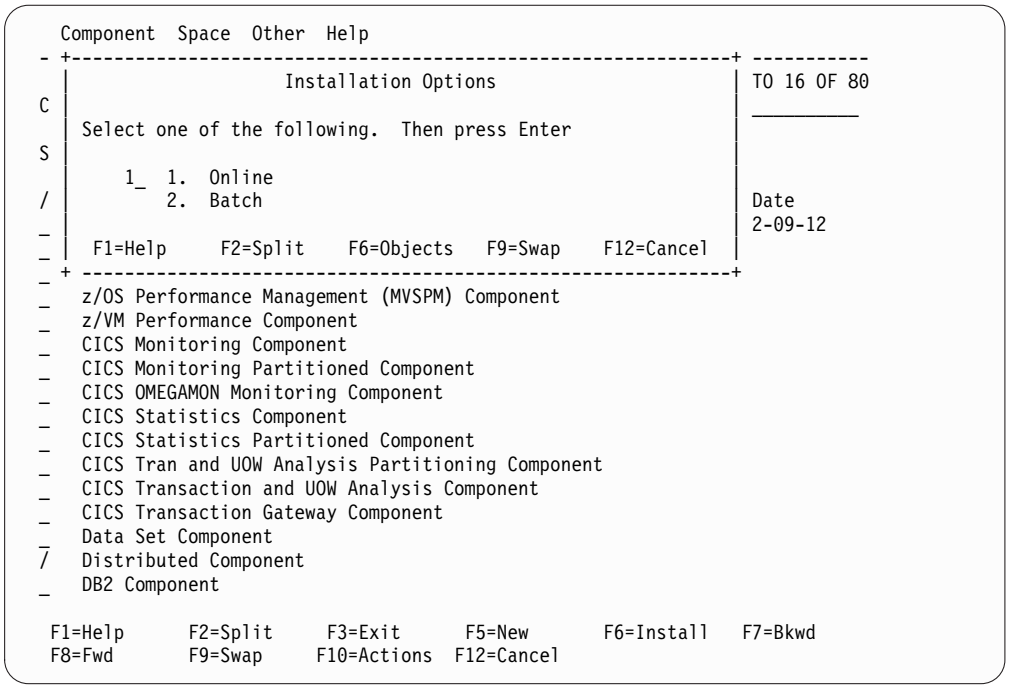


Figure 5. Installation Options window

4. Using the component-installation procedure in the *Administration Guide and Reference*, SH19-6816, specify whether the component is to be installed online or in batch mode.

   Batch mode installation results in less output than online mode. Furthermore, for online installation your terminal will be blocked for the duration of the installation. Therefore, it is recommended that you install components in batch.

Testing the installation

About this task

Before starting the daily use of the Distributed component, run a few tests to check that:

Procedure

1. The installation was successful.
   - Tivoli Decision Support for z/OS is collecting the correct data.
   - The data is being stored correctly.
   - The correct data is being used for the creation of reports.
2. Any relevant lookup tables contain appropriate values.

Results

Refer to the *Administration Guide and Reference*, SH19-6816 for steps involved in testing the component installation.

Putting the component into production

About this task

After you run the tests and verify that the installation is successful, you can put the Distributed component into production.

Figure 6 shows the daily steps involved in using Tivoli Decision Support for z/OS.

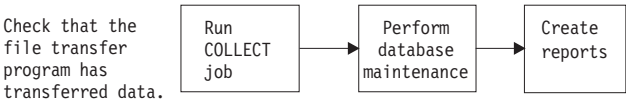


Figure 6. Daily steps involved in using Tivoli Decision Support for z/OS

You can run reports in batch, after setting batch parameters for each report using the administration dialog.

For detailed information about these steps, refer to the *Administration Guide and Reference*, SH19-6816.

Chapter 3. Installing and configuring the Windows component

This supplements the procedure in the *Administration Guide and Reference*, SH19-6816 for installing a component, with information specific to the Windows component.

The topic describes how to plan, install, and test the Windows component.

Planning the implementation process

About this task

Before installing the Windows component, you should follow these steps to plan the implementation process:

Procedure

1. Describe user tasks. Determine what data the Windows component must gather to help users accomplish those tasks.
2. Determine which Windows subcomponents you must install to meet the user needs.
3. Determine the administration tasks you must perform for the selected subcomponents, and make any decisions required by these tasks. These tasks help you customize Tivoli Decision Support for z/OS and the Windows component to work efficiently and effectively with your computer system.
4. Determine the tasks you must perform for each selected subcomponent when customizing the supported products to work with Tivoli Decision Support for z/OS and with the Windows component.

Results

If this is your first exercise in implementation planning follow all these t to ensure that the Windows component's implementation is consistent. If you are reading this topic in preparation for modifying your system, you might not need to perform all of these tasks.

Use the planning process to prepare for the following main customization tasks:

- Customizing Windows to generate the data required by the subcomponents you install.
- Defining environment data, which is all the information (in addition to the input data) that the Windows component needs to create reports. Environment data controls the data-collection process and provides more information in the reports.

Figure 7 on page 34 illustrates the process for implementing the Windows component.

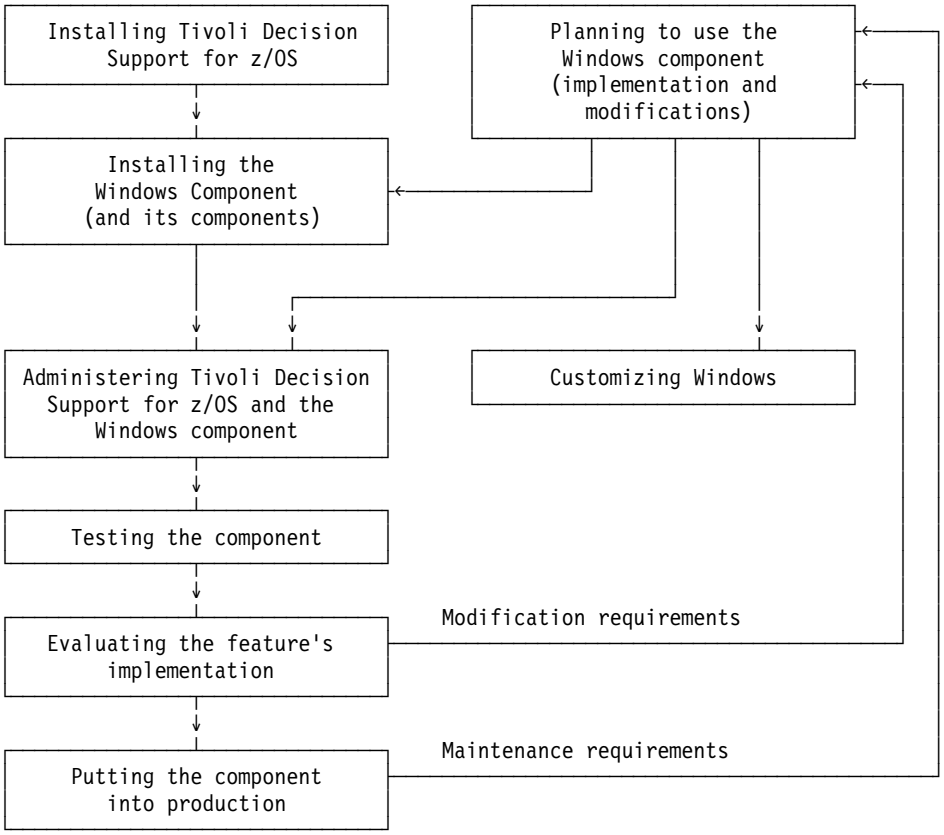


Figure 7. Implementation process for the Windows component

Considering which Windows subcomponents to install

Your most critical planning item is determining what information users need from the Windows component. For example, users may be interested only in processor information. Installing only those subcomponents needed to meet user requirements ensures that the feature benefits users while it minimizes the performance impact caused by data collection and interpretation activities.

The Windows component is divided into two subcomponents:
- Windows Performance
- Windows Device

Subcomponents are groups of Tivoli Decision Support for z/OS *objects*. For example, predefined update definitions, data tables, and reports. If you find that you need reports from a subcomponent that you have not installed, you must install that subcomponent and then wait until enough data is collected to create reports. However, if you install more subcomponents than you need, Tivoli Decision Support for z/OS collects needless data, which takes up disk space and uses processor time.

At this point, you might find it helpful to examine the predefined reports for each subcomponent, by turning to Chapter 6, “Reports,” on page 53.

Installing the Windows component on your Windows nodes

About this task

This section gives you step-by-step information on how to install Tivoli Decision Support for z/OS on your Windows nodes.

Step 1: Checking Windows requirements

About this task

Check the following Windows requirements:

- "Disk space requirements"
- "Software requirements"

Disk space requirements

About this task

You need the following approximate disk space on each Windows node:

Table 16. Disk space requirements on each Windows node

Installation task	Disk space (approximate)
Loading code in the file system. For example, `c:\tdszwin\bin`:	1MB
Creating configuration file in. For example, `c:\tdszwin\etc`:	1MB
Creating directory for data files in. For example, `c:\tdszwin\var`:	10MB (see note)
Note: The 10MB is based on the following assumptions: • Performance information written every 5 minutes (default) resulting in about 12 x 24 x 100 bytes (30K) per day. • Device information written every 30 minutes (default) resulting in about 2 x 24 x 100 bytes (5K) per day. • Allowance for a couple of hundred days of these files plus some overhead.	

Software requirements

About this task

The Windows component requires all of the following programs:

- Microsoft Windows 2003 Server or 2008 Server
- WMI enabled
- VBScript enabled

Step 2: Transferring Tivoli Decision Support for z/OS feature code to Windows

About this task

The Windows part of the Windows component is distributed in `DRL`*`xxx`*`.SDRLWS(DRLWIN)`. Download this zip file using these steps:

Procedure

1. Login to the Windows node and open a command window.

2. Enter the following commands (we will assume all the files are to be stored under `c:\tdszwin`, although you can place them anywhere as long as you adjust paths in later instructions: `bin` is for program binaries, `etc` for configuration and `var` for data files):

```
c:
mkdir \tdszwin
mkdir \tdszwin\bin
mkdir \tdszwin\etc
mkdir \tdszwin\var
cd \tdszwin\bin
```

3. Start an FTP session to your z/OS host, and receive the file from the z/OS user where you installed Tivoli Decision Support for z/OS. In the following example, the highlighted entries show the commands you must enter.

   yourmvs
   The IP address/name of the z/OS system where Tivoli Decision Support for z/OS was installed and where the tar file is located.

   epdmwin
   The TSO user ID that has read access to the tar file.

```
c:\tdszwin\bin> ftp yourmvs
Connected to yourmvs.yourcorp.com.
220-FTPSERVE at YOURMVS.SE.IBM.COM, 11:05:24 on 03/24/00
220 Connection will close if idle for more than 5 minutes.
Name (yourmvs:root): epdmwin
331 Send password please.
Password: secret
230 EPDMWIN is logged on.
ftp> binary
200 Representation type is IMAGE.
ftp> get 'drl181.sdrlws(drlwin)' epdmwin.zip
200 Port request OK.
125 Sending data set DRL181.SDRLWS(DRLWIN) FIXrecfm 128
10240 bytes received in 0.05 seconds (204.80 Kbytes/s)
ftp> quit
221 Quit command received. Goodbye.
c:\tdszwin\bin>
```

4. On the Windows node, use Windows Explorer to unzip the contents of epdmwin.zip into your `bin` directory.
5. (Optional) After completing the previous step you can remove the zip file. If you need the zip file in the future, it still exists on z/OS and can be used again.
6. A sample **`drl.cfg`** parameter file is distributed as`c:\tdszwin\bin\drl.cfg-sample`. Copy and rename the parameter file to`c:\tdszwin\etc\drl.cfg` using Windows Explorer.

Capturing Windows component data and transmitting to z/OS

About this task

This section describes the steps you must perform to set up the Windows component system to capture performance data, and to transmit the data to the z/OS system:

- Step 1: Customize Windows node.
- Step 2: Set up scheduled tasks.

Step 1: Customizing the Windows node

About this task

Edit the parameter file `c:\tdszwin\etc\drl.cfg` to change the default values supplied by the Windows component to those for your installation.

Parameter file values to change

```
NodeName     = IBM-BOX01
XmitServer   = pthbox3.au.ibm.com
XmitUser     = drluser
XmitPassword = drlpwd
DrlPerfdXlat = 'drldata.perfd.n{n5,5}.d{f12,6}'
```

Parameter
Description

NodeName
The unique identifier for this node (up to 16 characters).

XmitServer
Host name (TCP nodename or TCP internet address) of the z/OS host to receive the Windows component data files.

XmitUser
User name to use for transmitting to host.

XmitPassword
Password to use for transmitting to host. This `drl.cfg` file should be protected as much as practicable by the Windows operating system. In addition, the RACF user should be setup with only enough capabilities to write to the required locations.

DrlPerfdXlat
The translation string used to form the host file name. The characters in this translation string are copied exactly as they are, except when one of the following constructs is found (X and Y are considered to be numbers).

{nX,Y} Will cause characters from the NodeName to be copied to the file name. The X indicates start position (1 is the first character) and Y indicates the number of characters to copy.

{fX,Y} Will cause characters from the input file name to be copied to the file name. The X indicates start position (1 is the first character) and Y indicates the number of characters to copy.

Consequently, the node name of `"IBM-PAXPRIM"` and input file name`ofdrlperfd_20090529.txt`, the translation string `tdssrv.log.n{n5,7}.d{f12,6}` will result in the file`tdssrv.log.nPAXPRIM.d090529` being created on the host. Any non-valid characters in the resulting name will be replaced with `$`.

Step 2: Setting up scheduled tasks

About this task

The processes that will collect performance and disk space data during the day are initiated by Windows Scheduled Tasks.

The process which transmits the information to the host is also controlled by Windows Scheduled Tasks.

You can use other scheduling solutions, such as Tivoli Workload Scheduler, but they are not covered here.

In order to set up `drlperfd`, you should follow these steps(for Windows 2003 Server):

Procedure

1. From the **Start** menu, add a scheduled task under the control panel (the method to do this may vary depending on how your operating system is set up).
2. From the scheduled task wizard, click the **Next** button.
3. Click on **Browse**and select **`c:\tdszwin\bin\drlperfd.vbs`**, then click on the **Open** button.
4. Call the task **`drlperfd`**, choose **Daily** then click on the **Next** button.
5. Set the start time to **12:02 AM**, every day, starting today then click on the **Next** button.
6. Enter your workstation password twice then click on the **Next** button.
7. Check the **Open advanced properties** then click on the **Finish** button.
8. Insert **`wscript`** (note the space at end) before the current run text.
9. Append **`c:\tdszwin\etc\drl.cfg c:\tdszwin\var`** to the end of the run text.
   For example:
   ```
   wscript c:\tdszwin\bin\drlperfd.vbs c:\tdszwin\etc\drl.cfg c:\tdszwin\var
   ```
10. Click on the **OK** button and enter the password again, if necessary.
11. Exit from the scheduled task control panel application.

Results

Set up a similar task called `drlperfd_boot`, the only difference being that it should be scheduled to start when the computer starts.

Set up a task for `drlxmit.vbs` to start at 12:10AM, 12:20AM and 12:30AM.

From scheduled tasks, run the `drlperfd` task by right-clicking on it and selecting **Run**. You should find that the control file and log file are created in `c:\tdszwin\var`. Once a suitable time has passed (enough to reach a 5 minute boundary like 10:00, 10:05 or 12:35), the data file will also appear.

Installing Windows component on the z/OS system

About this task

When you install Windows component subcomponents, Tivoli Decision Support for z/OS will install the required log and record definitions, record procedures, and update definitions to Tivoli Decision Support for z/OS system tables. To install the Windows subcomponents, use the Administration dialog.

Perform steps 1 to 4, as follows:

Procedure

1. From the Tivoli Decision Support for z/OS Administration window (Figure 8 on page 39), select **2**, Components and press **Enter**.

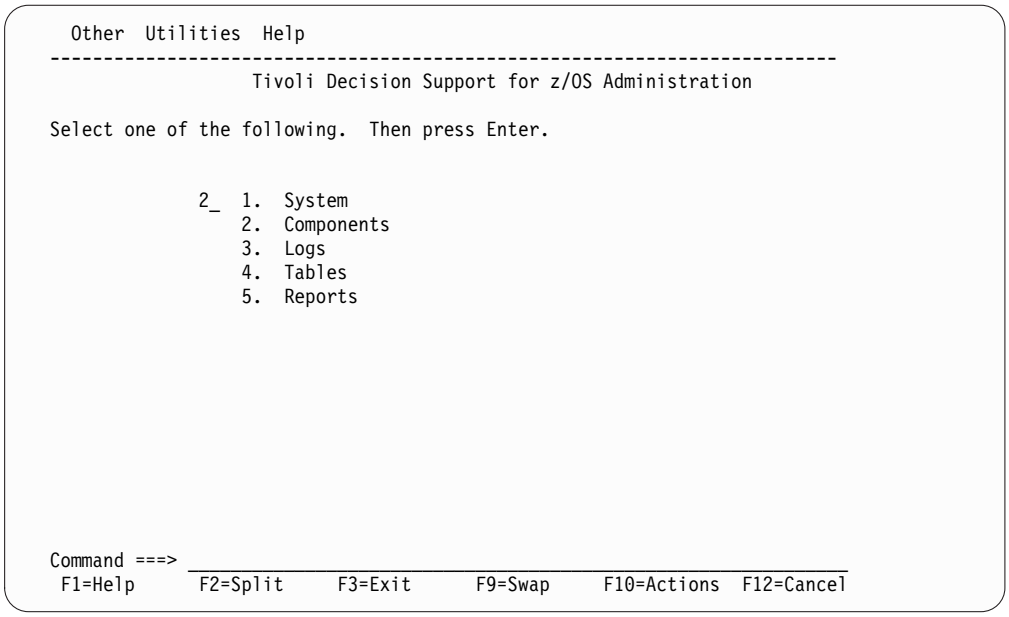


Figure 8. Tivoli Decision Support for z/OS Administration window

The Components window is displayed, (as shown in Figure 9).

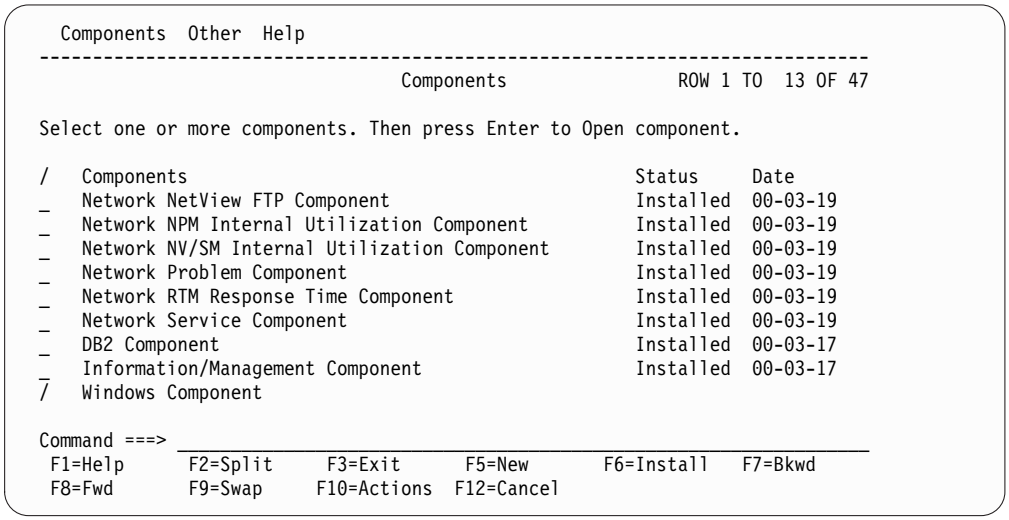


Figure 9. Components window

2. From the Components window, select the components to install (here, the **Windows component**) and press **F6**.
3. If individual subcomponents are listed, select the desired subcomponents for installation.
4. The Installation Options window is displayed, (as shown in Figure 10 on page 40).

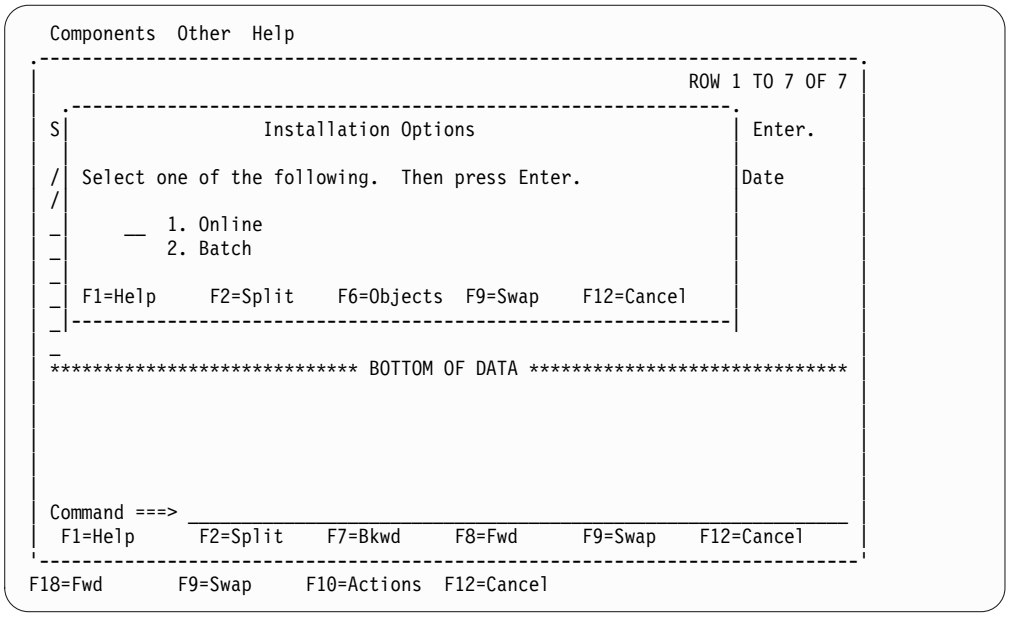


Figure 10. Installation Options window

5. Using the component-installation procedure in the*Administration Guide and Reference*, SH19-6816 specify whether the component is to be installed online, or in batch mode.

   Batch mode installation results in less output than online mode. Furthermore, for online installation your terminal will be blocked for the duration of the installation. Therefore, it is recommended that you install components in batch.

Collecting data under Tivoli Decision Support for z/OS

About this task

There are various methods you can define yourself to setup the JCL for the COLLECT procedure. This section, however, describes a method that uses a normal data set. It explains the JCL for a general **collect** job, which uses a data set to collect data into Tivoli Decision Support for z/OS tables.

General collect job

About this task

The member DRL*xxx*.SDRLCNTL(DRLJCOLL) contains sample JCL for a general collect.

Another way to obtain the COLLECT JCL, is to:

Procedure

1. Select 'Tivoli Decision Support for z/OS Administration' from the Tivoli Decision Support for z/OS Primary Menu.
2. Select '3. LOGS'.
3. Select 'Windows' from the list of logs, select the 'Utilities' pull-down, and press Enter.
4. Select '1. Collect'.
5. Type your installation-specific information in this window, and press Enter.

Results

Figure 11 shows an example of the general collect job.

```
//JOBCARD
//*                                                              *
//*  Notes:                                                      *
//*    Before you submit the job:                                *
//*    - Check the Tivoli Decision Support for z/OS and DB2   *
//*      data set names.                                         *
//*    - Check the DB2 subsystem name (default is DSN)           *
//*      and Tivoli Decision Support for z/OS                 *
//*      system table prefix (default is DRLSYS).                *
//*    - Insert the correct collect statement in DRLIN           *
//*      (as described above).                                   *
//*    - Specify the name of the log data set (or GDG) in        *
//*      DRLLOG. In this example, all existing generations       *
//*      are to be collected for files from Windows                 *
//*                                                              *
//****************************************************************
//COLLECT  EXEC PGM=DRLPLC,PARM=('SYSTEM=DB2A SYSPREFIX=DRLSYS')
//STEPLIB  DD DISP=SHR,DSN=DRL180.SDRLLOAD
//         DD DISP=SHR,DSN=db2loadlibrary
//DRLIN    DD *

COLLECT WINDOWS;

//DRLLOG   DD DISP=SHR,DSN=USER1.COLLECT.EPDMWIN
//DRLOUT   DD SYSOUT=*,DCB=(RECFM=F,LRECL=80)
//DRLDUMP  DD SYSOUT=*,DCB=(RECFM=F,LRECL=80)
/*
```

Figure 11. General COLLECT job

Testing the installation

About this task

Before starting the daily use of the Windows component, run a few tests to check that:

Procedure

1. The installation was successful.
   - Tivoli Decision Support for z/OS is collecting the correct data
   - The data is being stored correctly
   - The correct data is being used for the creation of reports
2. The lookup tables contain appropriate values.

Results

Refer to the *Administration Guide and Reference* for detail about the steps involved in testing component installation.

Putting the component into production

About this task

After you run the tests and verify that the installation is successful, you can put the Windows component and its subcomponents into production.

Figure 12 shows the daily steps involved in using Tivoli Decision Support for z/OS.

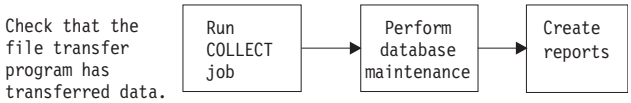


Figure 12. Daily steps involved in using Tivoli Decision Support for z/OS

You can run reports in batch, after setting batch parameters for each report using the administration dialog.

For detailed information about these steps, refer to the *Administration Guide and Reference*.

Part 2. Distributed Systems Performance Feature Reference

Chapter 4. Data flow and Tivoli Decision Support for z/OS objects

The chapters in this part describe:

- The general data flow, starting with the gathering of data at the distributed and Windows nodes into log files, and ending with the production of Tivoli Decision Support for z/OS reports.
- The data tables used for storing the collected information.
- The record specifications used for displaying information.
- The record definitions and log types used for mapping data.
- The data flow for these Distributed Systems Performance components, including the names of log files, and Tivoli Decision Support for z/OS records, tables, and reports.

Distributed Systems Performance feature general data flow

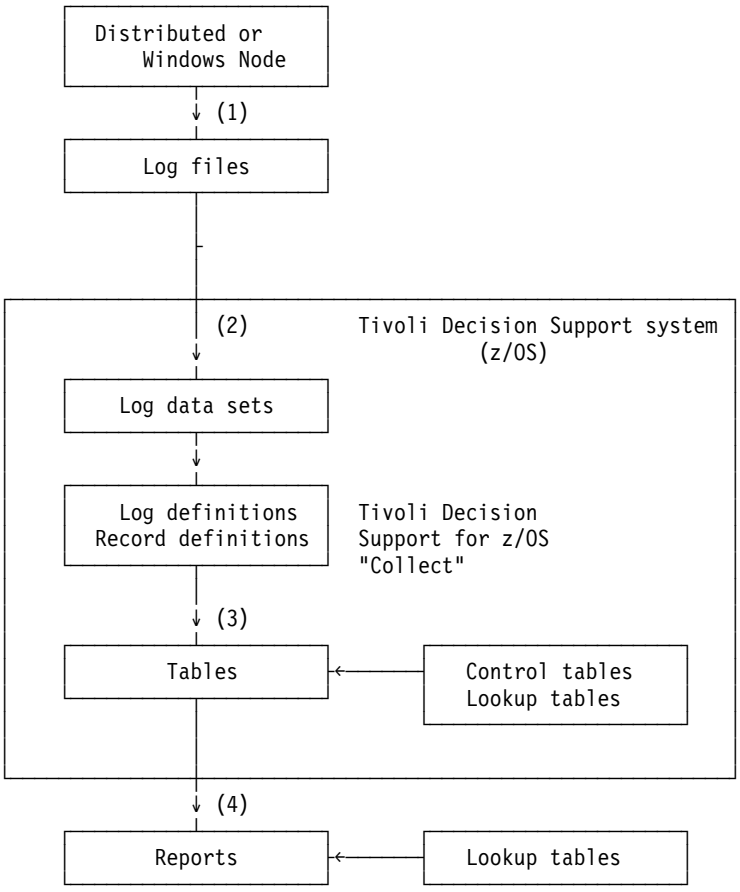


Figure 13. General Distributed Systems Performance feature data flow

The processing steps shown in Figure 13 can be viewed as:

1 Agents *gather* data into log files.

2 Those log file are transmitted to Tivoli Decision Support for z/OS as log data sets.

3 Tivoli Decision Support for z/OS *collects* those log data sets into tables.

4 Reports can be run to present the contents of those tables.

The following sections explain these steps in more detail:

1. **Gather data**, Step **1**

   The basic data used by the Distributed Systems Performance feature is *gathered* by agents into log files. These agents, which run on the distributed and Windows nodes, either issue commands or process files to gather the data.

2. **Transmit log file to Tivoli Decision Support for z/OS**, Step **2**

   TCP/IP is used to transmit the log files to Tivoli Decision Support for z/OS.

3. **Collect log data set information to Tivoli Decision Support for z/OS tables**, Step **3**

   In processing *each record contained in the log data set*, the collect procedure:

   a. Uses a *log collector* to update the Tivoli Decision Support for z/OS table with the record. To do this, the log collector:

      - Uses the log definition and record definitions to update the Tivoli Decision Support for z/OS table with the reformatted record.
      - Uses an *update definition* to decide which reformatted record fields are to be included in which Tivoli Decision Support for z/OS table, including further summarizing into other tables (for example, updating the monthly table DIST_DISK_M from the information used for updating the daily table DIST_DISK_D).
      - Takes information from *control tables*, where necessary.
      - Uses *lookup tables*, where necessary, to add user-defined data to the Tivoli Decision Support for z/OS table record.

   A description of the *collect* procedure is provided in the *Language Guide and Reference*, SH19-6817. A description of the use of control tables is provided in the *Administration Guide and Reference*, SH19-6816.

4. **Create reports**, Step **4**

   A description of how to create new reports is provided in the *Guide to Reporting*. The reports that can be created when using the Distributed Systems Performance feature are described in Chapter 6, “Reports,” on page 53.

Chapter 5. Data tables

The Tivoli Decision Support for z/OS database is a collection of DB2® tables, where each table contains a fixed number of columns. The number of rows in each table varies with time, because of rows added by the collect function and because of database maintenance.

This chaptertopic describes:

- The format that is used for defining Distributed Systems Performance feature table names and views
- The layout of the tables
- The tables used by the Distributed component
- The tables used by the Windows component

Note: For descriptions of common data tables used by the Distributed Systems Performance feature and other Tivoli Decision Support for z/OS features, refer to the *Administration Guide and Reference*, SH19-6816.

Naming standard for tables

The names of the Distributed Systems Performance feature tables use one of the following formats:

DIST_***content_suffix***
WIN_PERF_***content_suffix***

where:

- ***DIST*** or ***WIN_PERF*** specify whether the table is for the Distributed component or Windows component respectively.
- ***content*** is a description of the data held in the table (for example, DIST_***CPU***_D for the CPU statistics).
- ***suffix*** indicates the summarization level of the data in the table (for example, DIST_CPU_***D*** for CPU statistics summarized by day). A table name can have these summarization-level suffixes:
  _H The table holds data summarized by **hour** (hourly data).
  _D The table holds data summarized by **day** (daily data).
  _M The table holds data summarized by **month** (monthly data).

Table descriptions

Each table description includes information about the table, a description of each of the *primary key columns*, and a description of each of the *data columns*.

- Primary key columns are marked with a “K”. They are sorted in the sequence they appear in the key, from most significant to least.
- Data columns follow the last key column and are sorted in alphabetical order with the underscore ignored.

The descriptions of most key columns and data columns contain references to the fields from which they are derived in the record (for example, “From AC_UID”). For an explanation of such fields, refer to the log record definitions.

Tables with similar contents (that is, tables with the same name but with different suffixes) are described under one heading. For example, the heading "DIST_DISK_D, _M" covers two similar tables: DIST_DISK_D and DIST_DISK_M. Except for the DATE column, the contents of these tables are identical. Differences that exist in the contents of similar tables are explained in the column descriptions.

Tables in the Distributed component

This section describes the Distributed component tables:

DIST_CPU_H, _D, _M

These tables provide hourly, daily, and monthly statistics on utilization. They contain CPU and memory usage, paging rate and number of users and processes from distributed records with record type PERFVMA (see log record DIST_PERF_VM_A).

The default retention periods are:
7 days for DIST_CPU_H
30 days for DIST_CPU_D
765 days for DIST_CPU_M

Column name		Data type	Description
DATE	K	DATE	Date when the records were written. For DIST_CPU_M, this is the first day of the month. From DATE.
TIME	K	TIME	Time (rounded down to the nearest hour) when the record was written. From TIME. This field is present only in DIST_CPU_H.
NODE_NAME	K	CHAR(64)	Node name. From NODENAME.
CPU_USAGE_PCT_AVG		FLOAT	Average CPU usage, in percent. This is the average of CPUAVG.
CPU_USAGE_PCT_MAX		FLOAT	Maximum CPU usage, in percent. This is the maximum of CPUMAX.
CPU_USAGE_PCT_MIN		FLOAT	Minimum CPU usage, in percent. This is the minimum of CPUMIN.
MEASURED_SEC		INTEGER	Total measured time, in seconds. This is the sum of NUMSECS.
MEM_FREE_AVG		FLOAT	Average amount of free memory, in KB. This is the average of FREEAVG.
MEM_FREE_MAX		FLOAT	Maximum amount of free memory, in KB. This is the maximum of FREEMAX.
MEM_FREE_MIN		FLOAT	Minimum amount of free memory, in KB. This is the minimum of FREEMIN.
PAGE_RATE_AVG		FLOAT	Average number of pages paged IN/OUT per second. This is the average of PAGEAVG.
PAGE_RATE_MAX		FLOAT	Maximum number of pages paged IN/OUT per second. This is the maximum of PAGEMAX.

Column name		Data type	Description
PAGE_RATE_MIN		FLOAT	Minimum number of pages paged IN/OUT per second. This is the minimum of PAGEMIN.
PROCESSES_AVG		FLOAT	Average number of processes. This is the average of PROCS.
PROCESSES_MAX		INTEGER	Maximum number of processes. This is the maximum of PROCS.
PROCESSES_MIN		INTEGER	Minimum number of processes. This is the minimum of PROCS.
RECORDS_COLLECTED		INTEGER	Total number of records. This is the count of records.
USERS_AVG		FLOAT	Average number of users. This is the average of USERS.
USERS_MAX		INTEGER	Maximum number of users. This is the maximum of USERS.
USERS_MIN		INTEGER	Minimum number of users. This is the minimum of USERS.

DIST_DISK_D, _M

These tables provide daily and monthly statistics on disk usage. They contain disk performance data from records with record type DISKFSA (see log record DIST_DISK_FS_A).

The default retention periods are:
30 days for DIST_DISK_D
765 days for DIST_DISK_M

Column name		Data type	Description
DATE	K	DATE	Date when the records were written. For DIST_DISK_M, this is the first day of the month. From DATE.
NODE_NAME	K	CHAR(64)	Node name. From NODENAME.
DEVICE	K	CHAR(64)	Device name. From DEVNAME.
FILE_SYS	K	CHAR(64)	File system name. From FSNAME.
RECORDS_COLLECTED		INTEGER	Number of samples. This is the count of records.
SPACE_FREE_MB		FLOAT	Free space, in MB, accumulated for all samples. This is the sum of FREE. This value should be divided by RECORDS_COLLECTED to get the average free space.
SPACE_SIZE_MB		FLOAT	Total size of space, in MB, accumulated for all samples. This is the sum of SIZE. Divide this value by RECORDS_COLLECTED to get the average value.
SPACE_USED_PCT		FLOAT	Used space, in percent, accumulated for all samples. Calculated as the sum of 100*USED/SIZE. Divide this value by RECORDS_COLLECTED to get the average value.

DIST_DISKIO_H, _D

These tables provide hourly and daily statistics on disk I/O. They contain disk I/O performance data from records with record type PERFIOA (see log record DIST_PERF_IO_A).

The default retention periods are:
7 days for DIST_DISKIO_H
30 days for DIST_DISKIO_D

Column name		Data type	Description
DATE	K	DATE	Date when the records were written. From DATE.
TIME	K	TIME	Time (rounded down to the nearest hour) when the record was written. From TIME. This field is present only for DIST_DISKIO_H.
NODE_NAME	K	CHAR(64)	Node name. From NODENAME.
DISK_NAME	K	CHAR(64)	Physical disk name. From DEVICE.
DISK_BUSY_PCT_AVG		FLOAT	Average disk busy, in percent. This is the average of BUSYSUM.
DISK_BUSY_PCT_MAX		FLOAT	Maximum disk busy, in percent. This is the maximum of BUSYSUM.
MEASURED_SEC		INTEGER	Total measurement time, in seconds. Calculated as the sum of NUMSECS.
RDWR_KB_TOTAL		FLOAT	Amount of data read and written, in kilobytes. Calculated as the sum of RDWRSUM.
RDWR_MAXRATE		FLOAT	Maximum amount of data read and written, in kilobytes per second. Calculated as the sum of RDWRSUM/NUMSECS.
READ_KB_TOTAL		FLOAT	Amount of data read, in kilobytes. This is the sum of READSUM.
READ_MAXRATE		FLOAT	Maximum amount of data read, in kilobytes per second. Calculated as the maximum of READSUM/NUMSECS.
RECORDS_COLLECTED		INTEGER	Number of records. This is the count of records.
WRIT_KB_TOTAL		FLOAT	Amount of data written, in kilobytes. This is the sum of WRITSUM.
WRIT_MAXRATE		FLOAT	Maximum amount of data written, in kilobytes per second. Calculated as the maximum of WRITSUM/NUMSECS.

DIST_PAGING_H, _D, _M

These tables provide hourly, daily, and monthly statistics on page space usage. They contain page space performance data from records with record type PERFPSA (see log record DIST_PERF_PS_A).

The default retention periods are:
7 days for DIST_PAGING_H
30 days for DIST_PAGING_D
765 days for DIST_PAGING_M

Column name		Data type	Description
DATE	K	DATE	Date when the records were written. For DIST_PAGING_M, this is the first day of the month. From DATE.
TIME	K	TIME	Time (rounded down to the nearest hour) when the record was written. From TIME. This field is present only for DIST_PAGING_H.
NODE_NAME	K	CHAR(64)	Node name. From NODENAME
DISK_NAME	K	CHAR(64)	Physical disk name. From DISKNM.
PAGE_SPACE	K	CHAR(64)	Page space name. From PAGESPC.
PS_SIZE_AVG_MB		FLOAT	Average page space size, in MB. This is the average of SIZE.
PS_USED_AVG_MB		FLOAT	Average page space used, in MB. This is the average of USED.

Column name		Data type	Description
RECORDS_COLLECTED		INTEGER	Number of records. This is the count of records.

Tables in the Windows component

This section describes the Windows component tables:

- "WIN_PERF_DD_H, _D, _M"
- "WIN_PERF_PU_H, _D, _M"

WIN_PERF_DD_H, _D, _M

These tables holds hourly, daily and monthly data for the disk device records.

The default retention periods are:
10 days for WIN_PERF_DD_H
30 days for WIN_PERF_DD_D
765 days for WIN_PERF_DD_M

Column	Key	Datatype	Description
DATE	K	DATE	The date of the record. For the _M table, dates are stored as the first day of the month.
TIME	K	TIME	The time of the record. Set to the start of the hour for table _H. This column does not exist in the _D or _M tables.
NODE_NAME	K	CHAR(16)	The node name of the machine.
DEVICE_NAME	K	CHAR(8)	The device name for the device.
FREE_SPACE_AVG		FLOAT	Averages the free space on the device, based on SAMPLES.
FREE_SPACE_MAX		FLOAT	The maximum free space available.
FREE_SPACE_MIN		FLOAT	The minimum free space available.
PERCENT_FREE_AVG		FLOAT	Averages the free space percentage on the device, based on SAMPLES.
PERCENT_FREE_MAX		FLOAT	The maximum free space percentage.
PERCENT_FREE_MIN		FLOAT	The minimum free space percentage
SAMPLES		INTEGER	Count of the records consolidated into a table row. Used for working out averages.
TOTAL_DEVICE_SIZE		FLOAT	Device size at the end of the period.

WIN_PERF_PU_H, _D, _M

This table holds hourly, daily and monthly data for the performance usage records.

The default retention periods are:
7 days for WIN_PERF_PU_H
35 days for WIN_PERF_PU_D
366 days for WIN_PERF_PU_M

Column	Key	Datatype	Description
DATE	K	DATE	The date of the record. For the _M table, dates are forced back to the first day of the month.

Tables in the Windows component

Column	Key	Datatype	Description
TIME	K	TIME	The time of the record. Set to the start of the hour for table _H. This column does not exist in the _D or _M tables.
NODE_NAME	K	CHAR(16)	The node name of the machine.
CPU_COUNT		SMALLINT	Number of CPUs and/or cores.
CPU_USAGE_AVG		FLOAT	Averages the average CPU usage. This figure is averaged based on SAMPLES.
CPU_USAGE_MAX		FLOAT	The maximum CPU usage.
CPU_USAGE_MIN		FLOAT	The minimum CPU usage.
MEMORY_FREE_AVG		FLOAT	Averages the average free memory. This figure is averaged based on SAMPLES.
MEMORY_FREE_MAX		FLOAT	The maximum free memory.
MEMORY_FREE_MIN		FLOAT	The minimum free memory.
MEMORY_SIZE		FLOAT	The total memory size at the end of the period.
SAMPLES		INTEGER	Count of the records consolidated into a table row. Used for working out actuals.

Chapter 6. Reports

The reporting function produces reports based on the data in the Tivoli Decision Support for z/OS database. Reports can show data from tables or from views. You can request reports online or by submitting batch jobs. Typically, you use online reporting for reports that you use once, and batch reporting for regularly required reports.

This chaptertopic describes:

- The format of the names used to define each report, and how source tables, attributes and variables are used.
- The reports in the Distributed component
- The reports in the Windows component

Report format and general description

Tivoli Decision Support for z/OS presents reports in tables and graphs. All reports have the same basic report layout. This section describes the elements that are common among Tivoli Decision Support for z/OS feature reports:

- Report ID
- Report group
- Source
- Attributes
- Variables

Report ID

Tivoli Decision Support for z/OS assigns each report a unique identifier. The Distributed Systems Performance feature uses this format for report IDs:

component sequence

where:

component
: Identifies the component to which the report belongs. It can be one of the following:
DIST for the Distributed component.
WIN for the Windows component.

sequence
: A variable-length sequential number identifying the report.

Examples of report IDs are:

DIST04
WIN002

Report group

Tivoli Decision Support for z/OS uses several predefined report groups. For the Distributed Systems Performance feature, each component has one group. The report groups are DISTRIBUTED and WINDOWS for the Distributed and Windows components, respectively.

Source

Each report contains information from one or more source tables. The report descriptions in this chaptertopic list source tables. Refer to these source tables if you are interested in learning where certain data originates.

Attributes

Each report has certain attributes associated with it. Use these attributes as *keywords* to search for specific reports in the dialogs.

You can specify any number of attributes for a report, but the area to which the report belongs (for example, DISTRIBUTED) is always present for predefined reports.

You can also specify these attributes, when appropriate:

- Resource types, such as storage or processor
- Performance issues, such as availability or response
- Presentation forms, such as detail, overview, or trend
- Time resolutions, such as hourly, daily, or monthly

Variables

Each report has variables associated with it. You specify the values for these variables when you generate the report using the reporting dialog.

When you specify a date for a monthly report, specify the first day of the month. Otherwise, there is no match in the data table.

If a character variable happens to have only numeric characters, enclose it in single quote marks, otherwise it will not match the data. For example, if you have a system ID of 1234, specify it as '1234' on the Variables window.

Reports in the Distributed component

This section describes the Distributed component reports:

Distrib. CPU Util. for System, Hourly Trend

For a specific node in the network and a specific date, this graphical representation (see Figure 14 on page 55) shows the hourly trend of processor utilization over the entire day. Such information is useful as an entry point when investigating system performance.

This information identifies the report:

Report ID
DIST01

Report group
Distributed

Source
DIST_CPU_H (described on page "DIST_CPU_H, _D, _M" on page 48)

Attributes
Distributed, Performance, CPU, Utilization, Usage, Hourly, Trend

Variables
Date, Node name

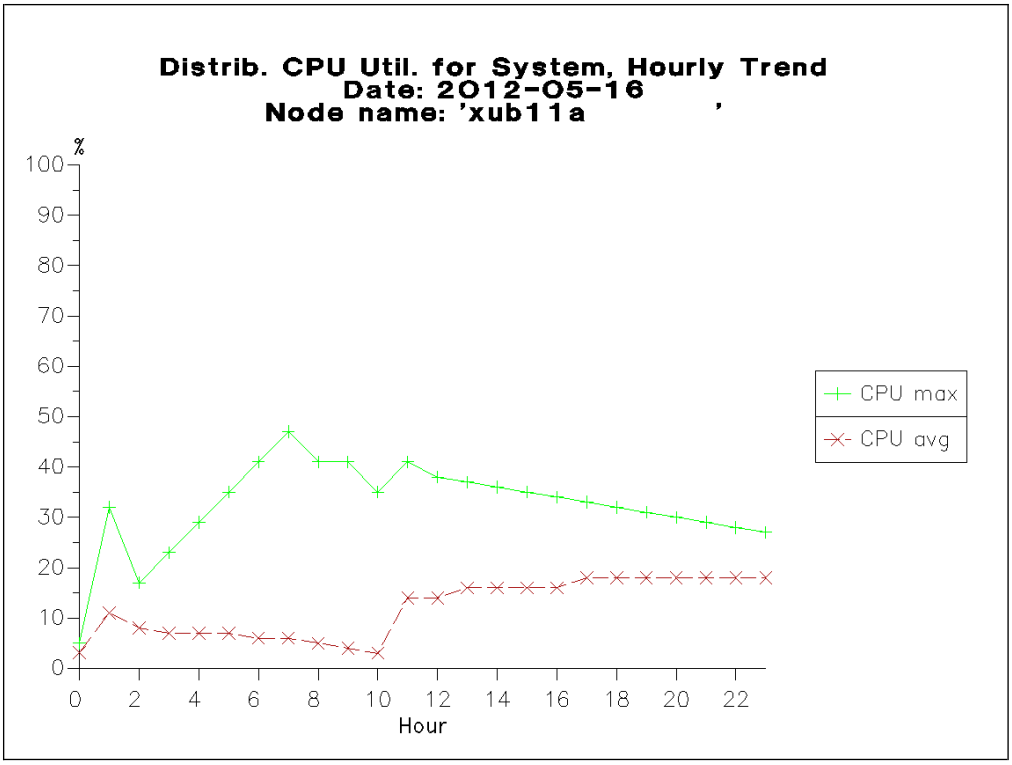


Figure 14. Example of Distrib. CPU Util. for System, Hourly Trend

The report contains this information:

Hour The hour of the day.

CPU avg
The average processor time that has been utilized, in percent.

CPU max
The maximum processor time that has been utilized, in percent.

Distrib. CPU Util. by System, Daily Overview

For all nodes in the network, this graphical representation (see Figure 15 on page 56) shows the average daily processor utilization. The report is produced by node name. Such information is useful as an entry point when investigating system performance. An hourly graphical display of processor utilization for a *specific node*, is given in "Distrib. CPU Util. for System, Hourly Trend" on page 54.

This information identifies the report:

Report ID
DIST02

Report group
Distributed

Source
DIST_CPU_D (described on page "DIST_CPU_H, _D, _M" on page 48)

Attributes
Distributed, Performance, CPU, Utilization, Usage, Daily, Overview

Variables
Date

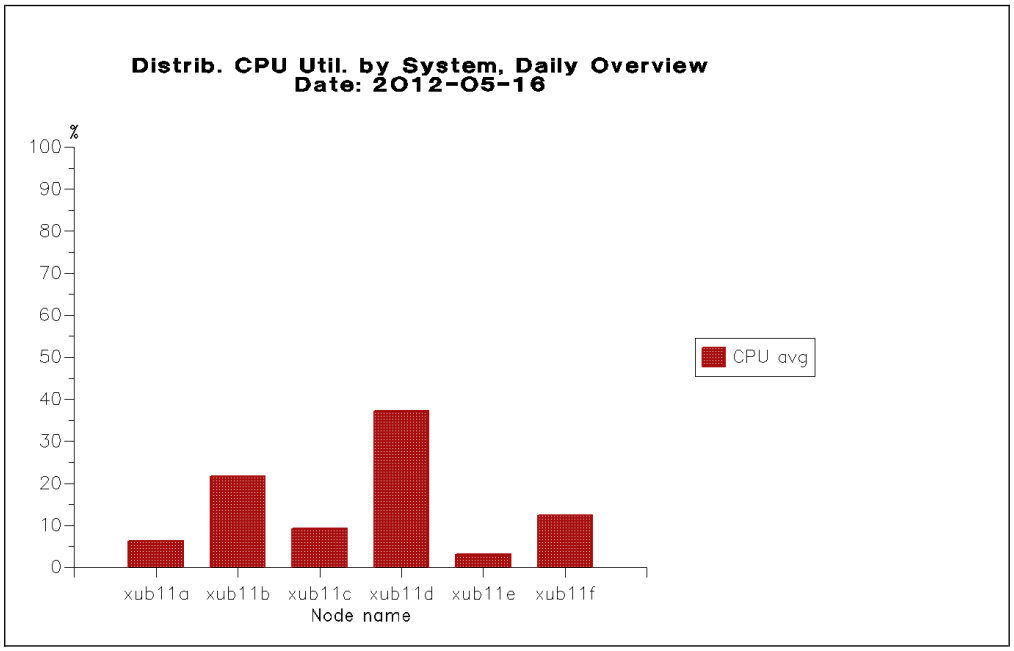


Figure 15. Example of Distrib. CPU Util. by System, Daily Overview

The report contains this information:

Node name
The name of the node in the network.

CPU avg
The average processor time that has been utilized, in percent.

Distrib. Statistics for System, Hourly Trend

For a specific node in the network, this report (see Figure 16 on page 57) provides hourly trend information about:

- Processor utilization (average and maximum).
- The amount of available memory (average and minimum).
- The paging rate (average and maximum).
- The number of processes (average and maximum).
- The number of users (average and maximum).

A total row is given, which are the averages, minimums or maximums calculated for all hours.

This information identifies the report:

Report ID
DIST03

Report group
Distributed

Source
DIST_CPU_H (described on page "DIST_CPU_H, _D, _M" on page 48)

Attributes
Distributed, Performance, System, Usage, Hourly, Trend

Variables
Date, Node name

Distrib. Statistics for System, Hourly Trend
Date: 2012-05-16
Node name: xub11a

Hour	CPU usage avg %	CPU usage max %	Mem free KB avg	Mem free KB min	Paging rate avg	Paging rate max	Procs avg	Procs max	Users avg	Users max
0	3.1	5	399103	398708	0.00	0	161.2	192	2.3	4
1	11.3	32	399103	398708	0.00	0	122.3	144	7.1	14
2	8.5	17	399103	398708	0.00	0	157.5	166	1.4	2
3	7.7	23	399103	398708	0.00	0	124.7	161	8.1	16
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
20	19.9	30	399103	398708	0.00	0	156.1	151	2.5	5
21	16.7	29	399103	398708	0.00	0	163.3	137	8.9	17
22	13.5	28	399103	398708	0.00	0	149.7	201	1.2	2
23	10.3	27	399103	398708	0.00	0	180.9	201	8.6	15
Total	11.9	47	399103	398708	0.00	0	161.0	161	4.2	9

Tivoli Decision Support: DIST03

Figure 16. Example of Distributed Statistics for System, Hourly Trend

The report contains this information:

Hour The hour of the day.

CPU usage avg %
The average processor utilization, in percent.

CPU usage max %
The maximum processor utilization, in percent.

Mem free KB avg
The average number of free kilobytes of memory.

Mem free KB min
The minimum number of free kilobytes of memory.

Paging rate avg
The average paging rate, per second.

Paging rate max
The maximum paging rate, per second.

Procs avg
The average number of processes.

Procs max
The maximum number of processes.

Users avg
The average number of users for this node.

Users max
The maximum number of users for this node.

Distrib. Statistics by System, Daily Overview

For all nodes in the network, this report (see Figure 17) provides daily overview information about:

- Processor utilization (average and maximum).
- The number of available pages of memory (average and minimum).
- The paging rate (average and maximum).
- The number of processes (average and maximum).
- The number of users (average and maximum).

An hourly breakdown of the information provided in this report, and for a specific node, is given in "Distrib. Statistics for System, Hourly Trend" on page 56.

This information identifies the report:

Report ID
DIST04

Report group
Distributed

Source
DIST_CPU_D (described on page "DIST_CPU_H, _D, _M" on page 48)

Attributes
Distributed, Performance, System, Usage, Daily, Overview

Variables
Date

Distrib. Statistics by System, Daily Overview
Date: 2012-05-16

Node name	CPU usage avg %	CPU usage max %	Mem free KB avg	Mem free KB min	Paging rate avg	Paging rate max	Processes avg	Processes max	Users avg	Users max
xub11a	6.2	41	399103	398708	0.00	0	141.9	172	17.4	38
xub11b	21.7	31	399103	398708	0.00	0	282.8	344	23.9	48
xub11c	9.3	12	399103	398708	0.00	0	223.7	316	16.5	36
xub11d	37.2	63	399103	398708	0.00	0	164.6	238	28.0	52
xub11e	3.1	15	399103	398708	0.00	0	105.5	200	19.6	36
xub11f	12.4	38	399103	398708	0.00	0	246.4	322	24.1	44

Tivoli Decision Support: DIST04

Figure 17. Example of Distrib. Statistics by System, Daily Overview

The report contains this information:

Node name
The name of the node in the network.

CPU usage avg %
The average processor utilization, in percent.

CPU usage max %
The maximum processor utilization, in percent.

Mem free KB avg
The average number of free kilobytes of memory.

Mem free KB min
The minimum number of free kilobytes of memory.

Paging rate avg
The average paging rate, per second.

Paging rate max
The maximum paging rate, per second.

Processes avg
The average number of processes.

Processes max
The maximum number of processes.

Users avg
The average number of users for this node.

Users max
The maximum number of users for this node.

Distrib. File Systems, Daily Overview

For a specific node in the network, this report (see Figure 18 on page 60) provides daily overview information about page space utilization: average allocated and free space (in megabytes), and the average percentage used space (by device and file system).

This information identifies the report:

Report ID
DIST05

Report group
Distributed

Source
DIST_DISK_D (described on page "DIST_DISK_D, _M" on page 49)

Attributes
Distributed, Performance, File, Utilization, Usage, Daily, Overview

Variables
Date, Node name

```
              Distrib. File systems, Daily Overview
                        Date: 2012-05-16
                       Node name: xub11a

                          | Space size  Space free  Space used
Device     File system    |   avg (MB)    avg (MB)     avg (%)
---------  -----------    | ----------  ----------  ----------
/dev/sda1  /              | 10000000.0   2470000.0        75.3
none       /dev           |   505348.0    504732.0         1.0
none       /dev/shm       |   512388.0    512276.0         1.0
none       /var/lock      |   512388.0    512388.0         0.0
none       /var/run       |   512388.0    512284.0         1.0

                 Tivoli Decision Support: DIST05
```

Figure 18. Example of Distrib. File Systems, Daily Overview

The report contains this information:

Device
The device used by the node.

File system
The file system corresponding to the given volume.

Space size avg (MB)
The average size of the space, in megabytes.

Space free avg (MB)
The average size of free space, in megabytes.

Space used avg (%)
The average size of used space, in percent.

Distrib. Disk I/O for Disk, Hourly Trend

For a specific node in the network and (optionally) a specific disk, this report (see Figure 19 on page 61) provides hourly trend information about the disk's utilization: the average and maximum percentage busy times, the average and maximum read amounts (in kilobytes per second), and the average and maximum write amounts (in kilobytes per second).

This information identifies the report:

Report ID
DIST06

Report group
Distributed

Source
DIST_DISKIO_H (described on page "DIST_DISKIO_H, _D" on page 49)

Attributes
Distributed, Performance, Disk, I/O, System, Hourly, Trend

Variables
Date, Node name, Disk

Distrib. Disk I/O for Disk, Hourly Trend
Date: 2012-05-16
Node name: xub11a
Disk: All

Hour	Busy avg (%)	Busy max (%)	Read avg (KB/sec)	Read max (KB/sec)	Write avg (KB/sec)	Write max (KB/sec)	RW avg (KB/sec)	RW max (KB/sec)
0	1.1	7.3	94.2	114.9	39.6	56.7	95.2	123.9
1	3.4	3.7	88.7	113.7	48.3	55.7	91.6	162.4
2	2.7	8.4	9.1	111.5	20.9	63.7	104.8	155.2
:	:	:	:	:	:	:	:	:
22	1.3	15.1	93.8	112.3	19.7	57.7	97.4	143.1
23	0.5	3.5	67.2	114.1	46.2	44.7	92.2	139.5
Avg/max	1.6	23.8	42.4	156.8	33.1	71.7	93.1	183.3

Tivoli Decision Support: DIST06

Figure 19. Example of Distrib. Disk I/O for Disk, Hourly Trend

The report contains this information:

Hour The hour of the day.

Busy avg (%)
The average time in which the processor was busy, in percent.

Busy max (%)
The maximum time in which the processor was busy, in percent.

Read avg (KB/sec)
The average amount of disk space read per second, in kilobytes.

Read max (KB/sec)
The maximum amount of disk space read per second, in kilobytes.

Write avg (KB/sec)
The average amount of disk space written per second, in kilobytes.

Write max (KB/sec)
The maximum amount of disk space written per second, in kilobytes.

RW avg (KB/sec)
The average amount of disk space read and written per second, in kilobytes.

RW max (KB/sec)
The maximum amount of disk space read and written per second, in kilobytes.

Distrib. Disk I/O for System, Daily Overview

For a specific node in the network, this report (see Figure 20 on page 62) provides daily overview information about the utilization of disks at the node: the average and maximum percentage busy times, the average and maximum read amounts (in kilobytes per second), and the average and maximum write amounts (in kilobytes per second). If you require more detailed hourly trend information about a disk's utilization, you can proceed to the report “Distrib. Disk I/O for Disk, Hourly Trend” on page 60.

This information identifies the report:

Report ID
DIST07

Report group
Distributed

Source
DIST_DISKIO_D (described on page "DIST_DISKIO_H, _D" on page 49)

Attributes
Distributed, Performance, Disk, I/O, System, Daily, Overview

Variables
Date, Node name

```
                          Distrib. Disk I/O for System, Daily Overview
                                       Date: 2012-05-16
                                       Node name: xub11a

           |  Busy avg  Busy max  Read avg  Read max  Write avg  Write max   RW avg    RW max
Disk name  |    (%)       (%)    (KB/sec)  (KB/sec)  (KB/sec)   (KB/sec)  (KB/sec)  (KB/sec)
---------  | --------  --------  --------  --------  ---------  ---------  --------  --------
sda        |      1.1       2.6       1.4       4.3        6.9        9.7       4.2      14.0
sdb        |      3.7       9.3       1.8       3.9        8.9        9.5       5.4      13.8
           | ========  ========  ========  ========  =========  =========  ========  ========
   Avg/max |      2.4       9.3       1.6       4.3        7.9        9.7       4.8      14.0

                                   Tivoli Decision Support: DIST07
```

Figure 20. Example of Distrib. Disk I/O for System, Daily Overview

The report contains this information:

Disk name
The name of the physical disk.

Busy avg (%)
The average time in which the disk was busy, in percent.

Busy max (%)
The maximum time in which the disk was busy, in percent.

Read avg (KB/sec)
The average amount of disk that was read per second, in kilobytes.

Read max (KB/sec)
The maximum amount of disk that was read per second, in kilobytes.

Write avg (KB/sec)
The average amount of disk that was written per second, in kilobytes.

Write max (KB/sec)
The maximum amount of disk that was written per second, in kilobytes.

RW avg (KB/sec)
The average amount of disk that was read and written per second, in kilobytes.

RW max (KB/sec)
The maximum amount of disk that was read and written per second, in kilobytes.

Distrib. Page Space Util., Hourly Trend

For a specific node in the network, this report (see Figure 21 on page 63) provides daily overview information about the utilization of page space at the node. For each page space, information about the disk name for the page space, page space size, and the amount of page space used (in megabytes and also as a percentage), is given.

This information identifies the report:

Report ID
DIST08

Report group
Distributed

Source
DIST_PAGING_H (described on page "DIST_PAGING_H, _D, _M" on page 50)

Attributes
Distributed, Performance, Page, Space, Utilization, Hourly, Trend

Variables
Date, Node name

```
          Distrib. Page space Util., Hourly Trend
                    Date: 2012-05-16
                    Node name: xub11a

      Page space              | Page space  Page space  Page space
Hour     name     Disk name   |  size (MB)   used (MB)   used (%)
----  ----------  ---------   | ----------  ----------  ----------
   0  /dev/sda15  /dev/sda15  |       2048           0         0.0
   1  /dev/sda15  /dev/sda15  |       2048        1024        50.0
   2  /dev/sda15  /dev/sda15  |       2048         512        25.0
   :      :           :       |          :           :          :
  22  /dev/sda15  /dev/sda15  |       2048        1536        75.0
  23  /dev/sda15  /dev/sda15  |       2048         256        12.5

              Tivoli Decision Support: DIST08
```

Figure 21. Example of Distrib. Page Space Util., Hourly Trend

The report contains this information:

Hour The hour of the day.

Page space name
The name of the page space used by the node.

Disk name
The physical disk name.

Page space size (MB)
The size of the page space, in megabytes.

Page space used (MB)
The amount of page space used, in megabytes.

Page space used (%)
The amount of page space used, in percent.

Distrib. CPU Util. Exceptions

For a specific node in the network, this report (see Figure 22 on page 64) provides exception reporting on average and maximum CPU usage:

- Node name.
- The table, date and time of the exception.
- The exception details (average and maximum CPU usage).

This information identifies the report:

Report ID
DIST09

Report group
Distributed

Source
DIST_CPU_H,_D,_M (described on page "DIST_CPU_H, _D, _M" on page 48)

Attributes
Distributed, Performance, CPU, Utilization, Usage, Exception

Variables
Start date, Average CPU reporting threshold, Maximum CPU reporting threshold

```
                    Distrib. CPU Util. Exceptions

                    Start date       : 2012-05-01
                    Average threshold: 999
                    Maximum threshold: 40

Node name   Type      Date         Hour   Average CPU   Maximum CPU
---------   -------   ----------   ----   -----------   -----------
xub11a      HOURLY    2012-05-16      6           6.0            41
                                      7           6.0            71
                                     11           4.0            48

                      2012-05-17      6           6.3            41
                                     11          14.0            55

xub11b      HOURLY    2012-05-16      7          12.1            48
                                     11          11.6            49

                      2012-05-17     11           4.0            52
                                     12           4.0            51

                     Tivoli Decision Support: DIST09
```

Figure 22. Example of Distributed CPU Utilization Exceptions

The report contains this information:

Node name
The node at which the exception occurred.

Type The source table in which the exception was found.

Date The date of the exception.

Hour The hour of the exception.

Average CPU
The average processor utilization, in percent. Either this column, or the next, or both, will excceed the relevant threshold.

Maximum CPU
The maximum processor utilization, in percent. Either this column, or the previous, or both, will excceed the relevant threshold.

Reports in the Windows component

This section describes the Windows component reports:

- "Windows Disk Usage for System, Hourly Trend" on page 65.
- "Windows Disk Usage for Device, Daily Overview" on page 65.
- "Windows CPU Utilization by System, Hourly Trend" on page 66.
- "Windows CPU Utilization by System, Daily Overview" on page 67.

- "Windows Memory Usage by System, Hourly Trend" on page 68.
- "Windows Memory Usage by System, Daily Overview" on page 69.
- "Windows System Overview Report" on page 70.

Windows Disk Usage for System, Hourly Trend

This report shows the hourly trend of disk usage on a selected days for selected nodes. It allows you to identify potential periods during the day when unusually high disk usage is taking place.

This information identifies the report:

Report ID
WIN001

Report group
WINDOWS

Source
WIN_PERF_DD_H

Attributes
WINDOWS, DISK, DEVICE, TREND, FREE, SPACE

Variables
FROM_DATE, TO_DATE, NODE_NAME

Windows Disk Usage for System, Hourly Trend
Date: '2009-06-12' to '2009-06-12'

NODE NAME	DATE	DEVICE NAME	TIME	FREE SPACE MIN (MB)	FREE SPACE AVG (MB)	FREE SPACE MAX (MB)	PERCENT FREE MIN	PERCENT FREE AVG	PERCENT FREE MAX
IBM-PAXPRIM	2009-06-12	C:	00.00.00	61691	61691	61691	67	67	67
		C:	01.00.00	61684	61688	61691	67	67	67
		C:	02.00.00	61684	61684	61684	67	67	67
		C:	03.00.00	61684	61684	61684	67	67	67
		C:	04.00.00	61684	61684	61684	67	67	67
		C:	05.00.00	61684	61702	61720	67	67	67
		C:	06.00.00	61720	61720	61720	67	67	67
		C:	07.00.00	61720	61720	61720	67	67	67
		C:	08.00.00	61720	61720	61720	67	67	67
		C:	09.00.00	61720	61720	61720	67	67	67
		C:	10.00.00	61594	61595	61595	67	67	67
		C:	11.00.00	61587	61588	61588	67	67	67
		C:	12.00.00	61586	61587	61587	67	67	67
		C:	13.00.00	61584	61586	61587	67	67	67
		C:	14.00.00	61565	61585	61604	67	67	67
		C:	15.00.00	61634	61635	61635	67	67	67
		C:	16.00.00	61631	61633	61634	67	67	67
		C:	17.00.00	61629	61633	61636	67	67	67
		C:	18.00.00	61635	61635	61635	67	67	67
		C:	19.00.00	61633	61634	61634	67	67	67
		C:	20.00.00	61632	61632	61632	67	67	67
		C:	21.00.00	61631	61632	61632	67	67	67
		C:	22.00.00	61630	61631	61631	67	67	67
		C:	23.00.00	61629	61630	61630	67	67	67

Tivoli Decision Support: WIN001

Figure 23. Example of Windows Disk Usage for System, Hourly Trend

Windows Disk Usage for Device, Daily Overview

This report shows the daily overview of disk usage for the selected nodes and devices, if any. It allows you to identify potential adverse changes in disk usage.

This information identifies the report:

Report ID
WIN002

Report group
WINDOWS

Source
WIN_PERF_DD_D

Attributes
WINDOWS, DISK, DEVICE, OVERVIEW, FREE, SPACE

Variables
NODE_NAME, DEVICE_NAME

Windows Disk Usage for Device, Daily Overview

DATE	NODE NAME	DEVICE NAME	FREE SPACE MIN (MB)	FREE SPACE AVG (MB)	FREE SPACE MAX (MB)	PERCENT FREE MIN	PERCENT FREE AVG	PERCENT FREE MAX
2009-06-12	IBM-PAXPRIM	C:	61565	61652	61720	67	67	67

Tivoli Decision Support: WIN002

Figure 24. Example of Windows Disk Usage for Device, Daily Overview

Windows CPU Utilization by System, Hourly Trend

This report shows the hourly trend of CPU usage for the selected dates and nodes. It allows you to identify potential problems with processor usage.

This information identifies the report:

Report ID
WIN003

Report group
WINDOWS

Source
WIN_PERF_PU_H

Attributes
WINDOWS, CPU, UTILIZATION, TREND, FREE

Variables
FROM_DATE, TO_DATE, NODE_NAME

Windows CPU Utilization by System, Hourly Trend
Date: '2009-04-15' to '2009-06-12'

NODE NAME	DATE	TIME	CPU USAGE MIN PERCENT	CPU USAGE AVG PERCENT	CPU USAGE MAX PERCENT
IBM-PAXPRIM	2009-06-12	00.00.00	4	4	7
		01.00.00	4	5	5
		02.00.00	5	5	5
		03.00.00	5	5	5
		04.00.00	4	5	5
		05.00.00	4	4	5
		06.00.00	4	4	4
		07.00.00	4	4	4
		08.00.00	4	4	4
		09.00.00	4	4	5
		10.00.00	4	5	5
		11.00.00	5	5	5
		12.00.00	5	5	5
		13.00.00	5	5	5
		14.00.00	5	5	5
		15.00.00	5	5	5
		16.00.00	5	5	5
		17.00.00	5	5	6
		18.00.00	6	6	6
		19.00.00	6	7	7
		20.00.00	7	7	8
		21.00.00	8	8	8
		22.00.00	8	9	9
		23.00.00	9	9	9

Tivoli Decision Support: WIN003

Figure 25. Example of Windows CPU Utilization by System, Hourly Trend

Windows CPU Utilization by System, Daily Overview

This report shows the daily overview of CPU usage for the selected dates and nodes. It allows you to identify potential problems with processor usage over the longer term.

This information identifies the report:

Report ID
WIN004

Report group
WINDOWS

Source
WIN_PERF_PU_D

Attributes
WINDOWS, CPU, UTILIZATION, OVERVIEW, FREE

Variables
NODE_NAME

Windows CPU Utilization by System, Daily Overview

NODE NAME	DATE	CPU USAGE MIN PERCENT	CPU USAGE AVG PERCENT	CPU USAGE MAX PERCENT
IBM-PAXPRIM	2009-06-12	4	5	9
IBM-PAXPRIM	2009-06-13	2	15	74
IBM-PAXPRIM	2009-06-14	1	16	79

Tivoli Decision Support: WIN004

Figure 26. Example of Windows CPU Utilization by System, Daily Overview

Windows Memory Usage by System, Hourly Trend

This report shows the hourly trend of memory usage for the selected dates and nodes. It allows you to identify potential problems with memory usage during the day.

This information identifies the report:

Report ID
WIN005

Report group
WINDOWS

Source
WIN_PERF_PU_H

Attributes
WINDOWS, MEMORY, USAGE, TREND, FREE

Variables
FROM_DATE, TO_DATE, NODE_NAME

Windows Memory Usage by System, Hourly Trend
Date: '2009-06-12' to '2009-06-12'

NODE NAME	DATE	TIME	MEMORY FREE MIN (KB)	MEMORY FREE AVG (KB)	MEMORY FREE MAX (KB)
IBM-PAXPRIM	2009-06-12	00.00.00	1399924	1407371	1408564
		01.00.00	1364320	1380610	1407532
		02.00.00	1367096	1367813	1368684
		03.00.00	1366520	1367241	1367988
		04.00.00	1366172	1366796	1367520
		05.00.00	1346256	1380571	1389388
		06.00.00	1387428	1388094	1388876
		07.00.00	1376300	1387421	1388156
		08.00.00	1386156	1386948	1387664
		09.00.00	1015916	1353313	1399148
		10.00.00	1045336	1155013	1214188
		11.00.00	1015476	1056817	1120244
		12.00.00	1100632	1152908	1213140
		13.00.00	1062504	1138608	1200420
		14.00.00	920532	984165	1096668
		15.00.00	893760	988563	1047896
		16.00.00	924324	983061	1050484
		17.00.00	937800	958081	963300
		18.00.00	927856	941137	944116
		19.00.00	938352	940428	941956
		20.00.00	934356	939315	940936
		21.00.00	919056	937112	939584
		22.00.00	928324	933017	935268
		23.00.00	925836	932618	933872

Tivoli Decision Support: WIN005

Figure 27. Example of Windows Memory Usage by System, Hourly Trend

Windows Memory Usage by System, Daily Overview

This report shows the daily overview of memory usage for the selected nodes. It allows you to identify potential problems with memory usage over the medium to long term.

This information identifies the report:

Report ID
WIN006

Report group
WINDOWS

Source
WIN_PERF_PU_D

Attributes
WINDOWS, MEMORY, USAGE, TREND, FREE

Variables
NODE_NAME

Windows Memory Usage by System, Daily Overview

NODE NAME	DATE	MEMORY FREE MIN (KB)	MEMORY FREE AVG (KB)	MEMORY FREE MAX (KB)
IBM-PAXPRIM	2009-06-12	893760	1159459	1408564

Tivoli Decision Support: WIN006

Figure 28. Example of Windows Memory Usage by System, Daily Overview

Windows System Overview Report

This report shows the system overview for a given data range and node list. It is intended to provide a high-level health check for all listed Windows nodes.

This information identifies the report:

Report ID
WIN007

Report group
WINDOWS

Source
WIN_PERF_PU_H, WIN_PERF_DD_H

Attributes
WINDOWS, SYSTEM, MEMORY, DISK, USAGE, TREND, OVERVIEW

Variables
FROM_DATE, TO_DATE, NODE_NAME

Windows System Overview Report
Date: '2009-04-15' to '2009-04-15'

NODE NAME	DATE	TIME	CPU USAGE MIN %	CPU USAGE AVG %	CPU USAGE MAX %	MEMORY FREE MIN (KB)	MEMORY FREE AVG (KB)	MEMORY FREE MAX (KB)	FREE SPACE MIN %	FREE SPACE AVG %	FREE SPACE MAX %	FREE SPACE MIN (MB)	FREE SPACE AVG (MB)	FREE SPACE MAX (MB)
IBM-PAXW2K3	2009-04-15	01.00.00	0	0	0	761436	767177	769700	6	6	6	71059	71059	71059
		02.00.00	0	0	0	762992	767326	768264	6	6	6	71059	71059	71059
		03.00.00	0	0	0	762488	767168	769344	6	6	6	71059	71059	71059
		04.00.00	0	0	0	763248	767604	769248	6	6	6	71059	71059	71059
		05.00.00	0	0	0	762536	767026	767936	6	6	6	71059	71059	71059
		06.00.00	0	0	0	750628	765030	768428	6	6	6	71059	71059	71059
		07.00.00	0	0	0	761988	766864	768684	6	6	6	71059	71059	71059
		08.00.00	0	0	0	761828	766428	767796	6	6	6	71059	71059	71059
		09.00.00	0	0	0	762660	766984	768732	6	6	6	71059	71059	71059
		10.00.00	0	0	0	761768	766314	767748	6	6	6	71059	71059	71059
		11.00.00	0	0	0	761848	767150	769156	6	6	6	71059	71059	71059
		12.00.00	0	0	0	763076	767396	769036	6	6	6	71059	71059	71059
		13.00.00	0	0	0	761764	766653	768648	6	6	6	71059	71059	71059
		14.00.00	0	0	0	762248	766901	768456	6	6	6	71059	71059	71059
		15.00.00	0	0	0	761992	766368	767444	6	6	6	71059	71059	71059
		16.00.00	0	0	0	759528	765585	767356	6	6	6	71059	71059	71059
		17.00.00	0	0	0	761256	765821	767588	6	6	6	71059	71059	71059
		18.00.00	0	0	0	760948	765652	767776	6	6	6	71059	71059	71059
		19.00.00	0	0	0	761504	765825	767204	6	6	6	71059	71059	71059
		20.00.00	0	0	0	760660	765440	767024	6	6	6	71059	71059	71059
		21.00.00	0	0	0	760556	765945	767820	6	6	6	71059	71059	71059
		22.00.00	0	0	0	761356	765693	767436	6	6	6	71059	71059	71059
		23.00.00	0	0	0	760828	766078	768404	6	6	6	71059	71059	71059

Tivoli Decision Support: WIN007

Figure 29. Example of Windows System Overview Report

Chapter 7. Log record definitions

This chapter describes the following record definitions:

Note that all log record definitions following have a two-byte length and two-byte filler at the start. This is a result of transmitting the file into a variable length data set on z/OS. The actual files generated on the distributed platforms start at the third field in each record definition.

DIST_PERF_VM_A

This performance record definition is used for selecting records with record type PERFVMA.

Main section

Field name	Offset	Length	Format	Description
RECLEN	0	2	BINARY	Record length
*	2	2	BINARY	Unused
RECTYPE	4	7	CHAR	Record type (literally PERFVMA)
NODENAME	11	16	CHAR	Node name
DATE	27	8	CHAR (YYYYMMDD)	Date
TIME	35	6	TIME (HHMMSS)	Time
NUMSECS	41	4	EXTERNAL INTEGER	Measured time
FREEMIN	45	9	EXTERNAL INTEGER	Free memory in KB, minimum
FREEAVG	54	11	EXTERNAL FLOAT	Free memory in KB, average
FREEMAX	65	9	EXTERNAL INTEGER	Free memory in KB, maximum
PAGEMIN	74	9	EXTERNAL INTEGER	Paging rate, minimum
PAGEAVG	83	11	EXTERNAL FLOAT	Paging rate, average
PAGEMAX	94	9	EXTERNAL INTEGER	Paging rate, maximum
CPUMIN	103	9	EXTERNAL INTEGER	CPU usage, minimum
CPUAVG	112	11	EXTERNAL FLOAT	CPU usage, average
CPUMAX	123	9	EXTERNAL INTEGER	CPU usage, maximum
USERS	132	9	EXTERNAL INTEGER	Number of users
PROCS	141	9	EXTERNAL INTEGER	Number of processes

DIST_DISK_FS_A

This disk space performance record definition is used for selecting records with record type DISKFSA.

Main section

Field name	Offset	Length	Format	Description
RECLEN	0	2	BINARY	Record length
*	2	2	BINARY	Unused
RECTYPE	4	7	CHAR	Record type (literally DISKFSA)
NODENAME	11	16	CHAR	Node name
DATE	27	8	CHAR (YYYYMMDD)	Date
TIME	35	6	TIME (HHMMSS)	Time
SIZE	41	9	EXTERNAL INTEGER	Total space, in MB
FREE	50	9	EXTERNAL INTEGER	Free space, in MB
PCTUSED	59	3	EXTERNAL INTEGER	Used space, in percent

Record section DEVICE

Offset 62
Length
Length of DEVLEN + 4

Field name	Offset	Length	Format	Description
DEVLEN	0	4	EXTERNAL INTEGER	Length of field DEVNAME
DEVNAME	4	*	CHAR(*)	Device name

Record section SPACE

Offset 66 + length of section DEVICE
Length
Length of FSLEN + 4

Field name	Offset	Length	Format	Description
FSLEN	0	4	EXTERNAL INTEGER	Length of section SPACE_NAME
FSNAME	4	*	CHAR(*)	Space name

DIST_PERF_IO_A

This disk I/O performance record definition is used for selecting records with record type PERFIOA.

Main section

Field name	Offset	Length	Format	Description
RECLEN	0	2	BINARY	Record length

Field name	Offset	Length	Format	Description
*	2	2	BINARY	Unused
RECTYPE	4	7	CHAR	Record type (literally PERFIOA)
NODENAME	11	16	CHAR	Node name
DATE	27	8	CHAR (YYYYMMDD)	Date
TIME	35	6	TIME (HHMMSS)	Time
NUMSECS	41	4	EXTERNAL INTEGER	Measurement interval
READMIN	45	9	EXTERNAL INTEGER	KB read, minimum
READAVG	54	11	EXTERNAL FLOAT	KB read, average
READMAX	65	9	EXTERNAL INTEGER	KB read, maximum
READSUM	74	9	EXTERNAL INTEGER	KB read, total
WRITMIN	83	9	EXTERNAL INTEGER	KB written, minimum
WRITAVG	92	11	EXTERNAL FLOAT	KB written, average
WRITMAX	103	9	EXTERNAL INTEGER	KB written, maximum
WRITSUM	112	9	EXTERNAL INTEGER	KB written, total
RDWRMIN	121	9	EXTERNAL INTEGER	KB read and written, minimum
RDWRAVG	130	11	EXTERNAL FLOAT	KB read and written, average
RDWRMAX	141	9	EXTERNAL INTEGER	KB read and written, maximum
RDWRSUM	150	9	EXTERNAL INTEGER	KB read and written, total
BUSYSUM	159	5	EXTERNAL FLOAT	Busy %
DEVICE	164	*	CHAR(*)	Physical disk name

DIST_PERF_PS_A

This paging space performance record definition is used for selecting records with record type PERFPSA.

Main section

Field name	Offset	Length	Format	Description
RECLEN	0	2	BINARY	Record length
*	2	2	BINARY	Unused
RECTYPE	4	7	CHAR	Record type (literally PERFPSA)
NODENAME	11	16	CHAR	Node name
DATE	27	8	CHAR (YYYYMMDD)	Date
TIME	35	6	TIME (HHMMSS)	Time
PAGESPC	41	20	CHAR	Page space name
DISKNM	61	20	CHAR	Physical disk name
SIZE	81	5	EXTERNAL INTEGER	Page space size, MB

Field name	Offset	Length	Format	Description
USED	86	5	EXTERNAL INTEGER	Page space used, MB

WIN_PERF_DD

This record gives the device information for each non-removable device on the machine. The information is gathered with the following WMI command:

```
select Name, Size, FreeSpace
    from Win32_LogicalDisk
    where DriveType = 3
```

This is both gathered and reported based on the configuration item xa_XPerfDiskFsRun. Note that there is a single space following each field.

Main section

Column	Position	Size	Description
Type	0	2	DD to indicate this is a disk device record.
Date	3	8	YYYYMMDD of the record.
Time	12	6	HHMMSS of the record.
Node name	19	16	Node name of the machine.
Device name	36	8	Device name.
Total size	45	9	Zero-padded device size in megabytes.
Free space	55	9	Zero-padded device free space in megabytes.
Free %	65	3	Zero-padded percentage free.

WIN_PERF_PU

This record gives the minimum, maximum and average figures for CPU percentage use and memory usage. The information is gathered with the following WMI commands:

```
select PercentProcessorTime, TimeStamp_100NS
    from Win32_PerfRawData_PerfOS_Processor
    where Name = '_Total'
select TotalPhysicalMemory
    from Win32_LogicalMemoryConfiguration
select AvailableKBytes
    from Win32_PerfFormattedData_PerfOS_Memory
```

This is gathered continuously and reported (aggregated) based on the configuration item xb_XPerfCpuRun. Note that there is a single space following each field.

Main section

Column	Posit ion	Size	Description
Type	0	2	PU to indicate this is a performance usage record.
Date	3	8	YYYYMMDD of the record.
Time	12	6	HHMMSS of the record.
Node name	19	16	Node name of the machine.

Column	Posit ion	Size	Description
CPU count	36	2	Number of CPUs in this system.
Average CPU usage	39	3	Zero-padded average CPU percentage usage for period. This is the average of all the snapshots in the period.
Minimum CPU usage	43	3	Zero-padded minimum CPU percentage usage for period. A snapshot CPU usage figure is calculated by sampling all of the CPUs and averaging their usage (so a 2-CPU system running at 70% and 30% gives 50%). These snapshots are collected as often as possible but only reported based on the xb_XPerfCpuRun configuration item. This is the minimum snapshot for the period.
Maximum CPU usage	47	3	Zero-padded maximum CPU percentage usage for period. This is the minimum snapshot for the period.
Memory size	51	9	Zero-padded total memory size at end of period, in kilobytes.
Average memory	61	9	Zero-padded average memory free for period, in kilobytes.
Minimum memory	71	9	Zero-padded minimum memory free for period, in kilobytes (1024 bytes).
Maximum memory	81	9	Zero-padded maximum memory free for period, in kilobytes.

Part 3. Distributed Systems Performance Feature Guide (heritage)

Chapter 8. Introducing the Distributed Systems Performance feature

IBM Tivoli Decision Support for z/OS (hereafter referred to as Tivoli Decision Support for z/OS) is a reporting system that collects performance data logged by computer systems, summarizes the data, and presents it in a variety of forms for use in systems management. Tivoli Decision Support for z/OS consists of a base product and several optional features.

The Distributed Systems Performance feature is a solution for environments where z/OS is used, and whereAIXAIX , HP-UX, Sun Solaris, Linux, and Windows operating systems are installed on nodes within your network. The performance information from the various nodes is transferred to a central z/OS site, where reporting and analysis are performed.

This topic describes how information is gathered and collected using the Distributed Systems Performance feature.

Gathering and collecting performance data

At the nodes where Distributed Systems Performance feature agents have been installed and from which data is to be used, the basic information used by the Distributed Systems Performance feature is *gathered* by the agents into log files containing accounting, configuration, error, and performance data. The source data used by the Distributed Systems Performance feature to create the log files and a description of the log files, is given here:

Source data
: **Description of created log files**

UNIX accounting
: Log files containing accounting data. To create these log files, you must activate accounting in each node that is to be included in the performance analysis. However, activating accounting can place up to 20 percent load on the system.

UNIX configuration
: A log file containing configuration data (for hardware and software).

UNIX error
: A log file containing error data. Error logging is normally available in all nodes where the Distributed Systems Performance feature is installed and, therefore, no action by you should be required to create these log files.

UNIX performance and disk
: Log files containing processor use, I/O rate, and paging space data. These are continuously updated by a program supplied with the Distributed Systems Performance feature.

Windows performance and disk
: Log files containing processor and disk use. These are continuously updated by a program supplied with the Distributed Systems Performance feature.

The information contained in the log files must be transmitted to z/OS for use as data sets in the Tivoli Decision Support for z/OS COLLECT procedure. The most convenient way is to use a product such as TCP/IP. The `ftp` function of TCP/IP can be used to automate logon and file transfer.

After the transmission of log files is completed, the information contained in log data sets at the z/OS host is *collected* into Tivoli Decision Support for z/OS tables.

The collected data is combined with more data (called *environment data*) and is finally presented in reports.

The process of entering and maintaining environment data is called *administration*. Tivoli Decision Support for z/OS provides an administration dialog for maintaining resource information. Refer to the *Administration Guide and Reference*, SH19-6816 for information on how to use the administration dialog.

Figure 30 illustrates how data is organized for presentation in Tivoli Decision Support for z/OS reports.

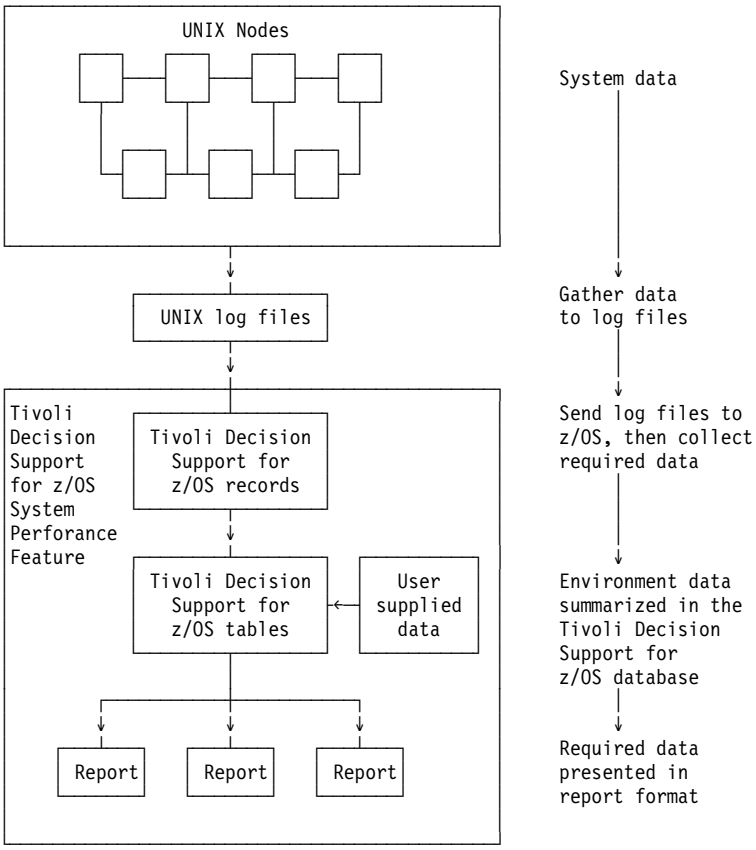


Figure 30. Organizing and presenting system performance data

Report groups

The reports produced by the Distributed Systems Performance feature are grouped in the following report groups:

XACCT
UNIX accounting reports

XCONFIG
UNIX configuration reports

XERROR
UNIX error reports

XPERF
UNIX performance reports

WINDOWS
Windows reports

The reports cover a wide range of needs in a data processing center, and reporting can be done online or in batch. They are accessible from the Tivoli Decision Support for z/OS reporting dialog.

Finally, the key to successful implementation of Tivoli Decision Support for z/OS is knowing:

- The information and resources on which you want to report and how to perform customization to select them
- The way you want to organize, set objectives for, and process the data (used later to define the environment)

Chapter 9. Installing and configuring the UNIX and Linux Performance components

This supplements the procedure in the *Administration Guide and Reference and Reference* for installing a component, with information specific to the UNIX Performance component.

Installing Linux on zSeries: For information about installing Linux on zSeries, see "Making input data available"in the *Tivoli Decision Support for z/OS System Performance Feature Reference Volume II*.

This topic describes how to plan, install, and test the UNIX Performance component and the Linux Performance component.

Planning the implementation process

About this task

Before installing the UNIX Performance component, you should follow these steps to plan the implementation process:

Procedure

1. Describe user tasks. Then determine what data the UNIX Performance component must gather to help users accomplish those tasks.
2. Determine which UNIX Performance subcomponent you must install to meet the user needs.
3. Determine the administration tasks you must perform for the selected subcomponents, and make any decisions required by these tasks. These tasks help you customize Tivoli Decision Support for z/OS and the UNIX Performance component to work efficiently and effectively with your computer system.
4. Determine (for each selected subcomponent) the tasks you must perform to customize the supported products to work with Tivoli Decision Support for z/OS and with the UNIX Performance component.

Results

If this is your first exercise in implementation planning, *follow all these steps* to ensure that the UNIX Performance component's implementation is consistent. If you are reading this topic in preparation for modifying your system, you might not need to perform all of these tasks.

Use the planning process to prepare for these main customization tasks:

- Customizing UNIX to generate the data required by the subcomponents you install.
- Defining environment data, which is all the information (in addition to the input data) that the UNIX Performance component needs to create reports. Environment data controls the data-collection process and provides more information in the reports.

Figure 31 illustrates the process for implementing the UNIX Performance component.

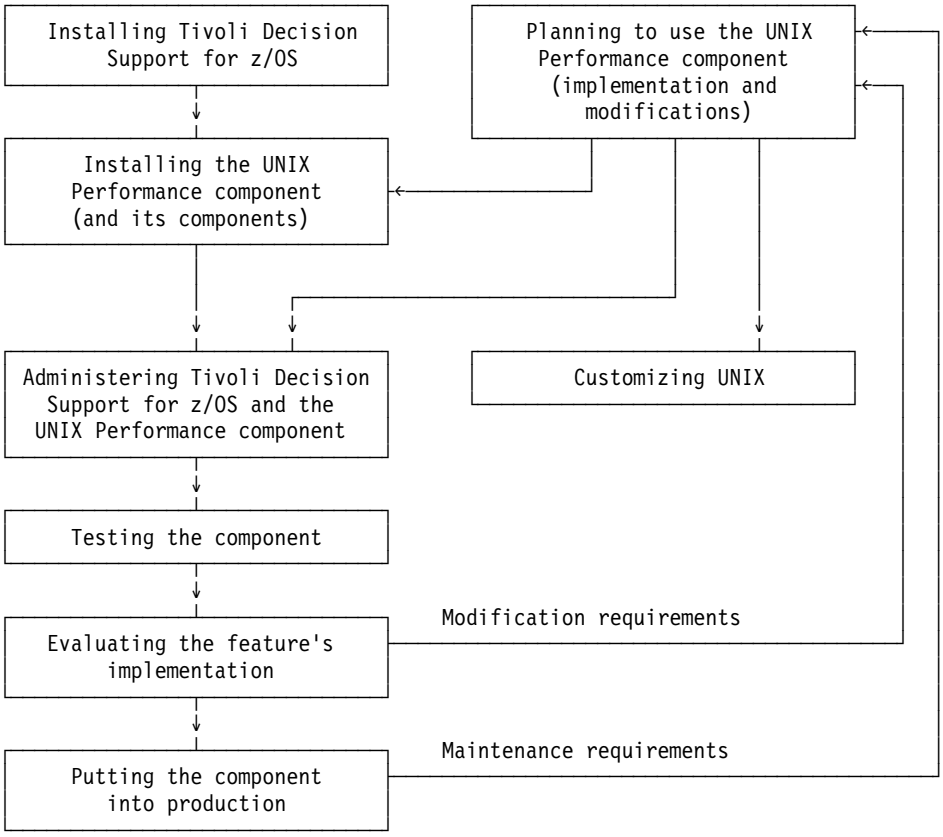


Figure 31. Implementation process for the UNIX Performance component

Considering which UNIX subcomponents to install

Your most critical planning task is determining what information users need from the UNIX Performance component. For example, users may be interested only in error conditions or in processor capacity. Installing only those subcomponents needed to meet user requirements ensures that the feature benefits users while it minimizes the performance impact caused by data collection and interpretation activities.

The UNIX Performance component is divided into four subcomponents:
- Accounting
- Configuration (AIX only)
- Error (AIX only)
- Performance

Subcomponents are groups of Tivoli Decision Support for z/OS *objects* (for example, predefined update definitions, data tables, and reports). If you find that you need reports from a subcomponent that you have not installed, you must install that subcomponent and then wait several days or weeks until enough data is collected to create reports. However, if you install more subcomponents than you need, Tivoli Decision Support for z/OS collects needless data, which takes up disk space and uses processor time.

Note: You should be aware that using the *process accounting* part of of the accounting subcomponent (log type PACCT) *will* degrade machine performance. However, using the other parts of the accounting subcomponent (log types QACCT, DTMP, and WTMP) *will not* degrade machine performance.

At this point, you might find it helpful to examine the predefined reports for each subcomponent, by turning to Chapter 14, "Reports," on page 151.

Installing the UNIX Performance component on your UNIX nodes

This section gives you practical step-by-step information on how to install Tivoli Decision Support for z/OS on your UNIX nodes.

Step 1: Check UNIX requirements

About this task

Check the following UNIX requirements:

- "Disk space requirements"
- "Software requirements"

Disk space requirements

You need the following approximate disk space on each UNIX node:

Table 17. Disk space requirements on each UNIX node

Installation task	Disk space (approximate)
Loading code in the file system /usr/lpp:	400KB
Creating work files for the volume group rootvg file system. For each node from which you are collecting data, and for all subcomponents, you require:	43MB for 10 users (see table note)
Note: The 43MB consists of: **Component** **Space required** **Accounting** 42.25MB (of which process accounting uses 42MB) **Performance** 0.4MB **Configuration** 180KB **Error** 3KB	

Software requirements

The UNIX Performance component requires one of the following programs:

- AIX 5L™ for Power V5.x

  Accounting Services (if you need to run the UNIX Resource Accounting)
- AIX 6 for Power V6.1
- HP-UX 11iv3 or earlier
- Sun Solaris 10 or earlier

Notes:

1. For AIX, HP, and Sun, you also require the corresponding communication programs to transfer log files to OS/390.

2. The Unix host name can be 16 characters long at maximum on AIX and 8 characters at maximum on all other platforms. If it is longer, it will be truncated.

Step 2: Transfer Tivoli Decision Support for z/OS feature code to UNIX

About this task

According to the operating system you are using, the UNIX part of the UNIX Performance component is distributed in the following SMP target library members:

AIX DRL*xxx*.SDRLWS(DRLAIX)

HP-UX
DRL*xxx*.SDRLWS(DRLHP11)

Sun Solaris
DRL*xxx*.SDRLWS(DRLSOLAR)

Download these `tar` files using these steps:

Procedure

1. Log in as root user on a UNIX node.
2. Enter the following commands:

   ```
   mkdir /usr/lpp/epdm
   cd /usr/lpp/epdm
   ```

3. Start an `ftp` session to your OS/390 host, and receive the file from the OS/390 user where you installed Tivoli Decision Support for z/OS. In the following example, the highlighted entries show the commands

   yourmvs
   is the IP address/name of the OS/390 system where Tivoli Decision Support for z/OS was installed and where the tar file is located.

   epdmunix
   is the TSO userid that has read access to the tar file.

   ```
   # ftp yourmvs
   Connected to yourmvs.yourcorp.com.
   220-FTPSERVE at YOURMVS.SE.IBM.COM, 11:05:24 on 03/24/00
   220 Connection will close if idle for more than 5 minutes.
   Name (yourmvs:root): epdmunix
   331 Send password please.
   Password: secret
   230 EPDMUNIX is logged on.
   ftp> binary
   200 Representation type is IMAGE.
   ftp> get 'drl181.sdrlws(drlaix)' epdmaix.tar
   ftp> get 'drl181.sdrlws(drlhp11)' epdmhp.tar
   ftp> get 'drl181.sdrlws(drlsolar)' epdmsolaris.tar
   200 Port request OK.
   125 Sending data set DRL181.SDRLWS(DRLAIX) FIXrecfm 128
   125 Sending data set DRL181.SDRLWS(DRLHP11) FIXrecfm 128
   125 Sending data set DRL181.SDRLWS(DRLSOLAR) FIXrecfm 128
   389120 bytes received in 25.2 seconds (6.349 Kbytes/s)
   ftp> quit
   221 Quit command received. Goodbye.
   #
   ```

4. When the UNIX Performance component code has been received by one UNIX node, send the tar files to all other UNIX nodes where the UNIX Performance component is to be installed.

5. On each of the UNIX nodes to which the tar files have been sent, enter the following tar command to unpack the epdm*xxxx* tar files (where *xxxx* is one of aix, hp, or solaris) and create the files in the directories `tar -xvf epdmxxxx.tar`
6. (Optional) After completing the previous step you can remove the tar files using the command `rm epdmxxxx.tar` (If you need the tar files in the future, they still exist on OS/390 and can be used again).
7. Create file system or directory for log files.
   - **For the AIX operating system**, the command `drlcrfs` is provided to create a file system with the name /var/epdm/, in volume group rootvg. The size of the file system is determined by an argument used with the `drlcrfs`. The UNIX Performance component uses the file system /var/epdm/ to store work and data files until they have been sent to OS/390.

     The `drlcrfs` command is also used to mount the file system using /var/epdm as the mount point.

     By having the /var/epdm/ directory in a separate file system, there is no risk of obtaining a "file system full" condition on file system /var (which would stop UNIX processing).

     For example, to create and mount a new file system with size 50MB, enter:

     ```
     /usr/lpp/epdm/bin/drlcrfs 50
     ```

     Confirm that the new file system has been created, by entering the following command:

     ```
     df /var/epdm
     ```
   - **For the HP and Sun operating systems**, the command `drlcrfs` is provided to create a directory with the name /var/epdm. The UNIX Performance component uses the directory /var/epdm to store work and data files until they have been sent to OS/390.

     For example, to create a directory, enter:

     ```
     /usr/lpp/epdm/bin/drlcrfs
     ```

     Confirm that the new directory has been created, by entering the following command:

     ```
     ls /var/epdm
     ```
8. Create symbolic links.

   To create these symbolic links, enter the command `/usr/lpp/epdm/bin/drlmklinks`

   Confirm that the symbolic links are created, using the following command:
9. The drl.cfg parameter file is distributed as /usr/lpp/epdm/etc/drl.cfg. Copy the parameter file to /etc/drl.cfg by entering `cp /usr/lpp/epdm/etc/drl.cfg /etc/drl.cfg`.

Results

Table 18 shows the files created by Steps 2 to 9.

Table 18. Files created on each system by Steps 2 to 9

Directory	Description	Created files:
/usr/lpp/epdm/bin/	This directory was created when the `tar` command was run. It contains executable UNIX Performance component commands.	drlpacct drlwtmp drldtmp drlqacct (see Note 1) drlerrpt (see Note 1) drlconfhw (see Note 1) drlconfsw (see Note 1) drlgather drlmklinks drlrmlinks drlperfd drliostat drlvmstat drlcrfs drlrmfs drlsend (see Note 2) drlpsperfd (see Note 2) drlactperfd (see Note 2) drldeactperfd (see Note 2) drlyday drlparmget (see Note 2) zccmsgwrite (see Note 2)
/usr/lpp/epdm/etc/	This directory was created when the `tar` command was run. It contains various files used by the UNIX Performance component.	drlcron drl.cfg (see Note 3) rc.epdmunix
/usr/lpp/epdm/lib/	This directory was created when the `tar` command was run. It contains libraries used by UNIX Performance component.	libDrl.a
/usr/lpp/epdm/lib/nls/msg/$LANG/	This directory was created when the `tar` command was run. It contains the message catalog used by UNIX Performance component.	drl.cat

Notes:

1. AIX only
2. A symbolic link has been created from /usr/bin/ to this program.
3. This file has been copied to /etc/.

Capturing UNIX Performance component data and transmitting to OS/390

About this task

This section describes:

- The steps you must perform to set up the UNIX Performance component system to capture performance data, and to transmit the data to the OS/390 system:
  Step 1: Set up UNIX accounting (accounting subcomponent).
  Step 2: Create log files
  Step 3: Define the receiver of UNIX Performance component data
  Step 4: Customize UNIX Performance component parameters
  Step 5: Start the drlperfd daemon (performance subcomponent)
  Step 6: Invoke drlperfd at start time (performance subcomponent)

  Note: Where a step is for a specific subcomponent only, the subcomponent name is given in parentheses in the heading for the step. Otherwise, you must carry out the step for all the subcomponents (accounting, configuration, error, and performance).
- The daily tasks that the UNIX Performance component will perform on your UNIX node if you use the default options when setting up the UNIX Performance component system to capture performance data. For more information see ("Description of daily tasks performed by UNIX Performance component" on page 99).

Step 1: Set up UNIX accounting

(**accounting subcomponent only**)

This section describes how to set up accounting for:

- AIX
- HP (on page "Setting up accounting for HP" on page 93)
- Sun (on page "Setting up accounting for Sun Solaris" on page 94)

Setting up accounting for AIX

About this task

This information can also be found in the section on setting up an accounting system in the *AIX System Management Guide*.

Procedure

1. Log in as root user.
2. Enter the `nulladm` command to ensure that the files have the correct access permissions, as follows: `/usr/sbin/acct/nulladm wtmp pacct`.
3. Update the `/usr/lib/acct/holidays` file. The UNIX Performance component does not use this file, because the accounting data is collected every hour, every day of the year. However, the accounting routines must have a valid file with a valid year in order to run correctly.

   Note: To define prime time, fill in the fields on the first data line (the first line that is not a comment), using a 24-hour clock. This line consists of three 4-digit fields, in the following order:
   a. Current year
   b. Beginning of prime time (*hhmm*)
   c. End of prime time (*hhmm*)

   Leading blanks are ignored. You can enter midnight as either 0000 or 2400. For example, to specify the year 2004, with prime time beginning at 8:00 a.m. and ending at 5:00 p.m., enter:

   ```
   2004  0800  1700
   ```

To define the company holidays for the year, fill in the fields on the next data line. This line consists of four fields, in the following order:

a. Day of the year
b. Month
c. Day of the month
d. Description of holiday

The day-of-the-year field must be a number from 1 through 365 (366 on leap year). It corresponds to the number of day on which the holiday falls. For example, February 1st is day 32. The other three fields are for information only and are treated as comments. A two-line example follows:

```
  1  Jan  1  New Year's Day
332  Nov 28  Thanksgiving Day
```

4. Turn on process accounting at start time by adding the following line to the `/etc/rc` file or by deleting the comment symbol (#) in front of the line if it exists:

   ```
   /usr/bin/su - adm -c /usr/sbin/acct/startup
   ```

   The startup procedure records the time that accounting was turned on and cleans up the previous day's accounting files.

5. Identify each file system that you want to include in disk accounting by adding the following line to the stanza for the file system in the `/etc/filesystems` file:

   ```
   account = true
   ```

   Example:

   ```
   /home:
           dev             = /dev/hd1
           vfs             = jfs
           log             = /dev/hd8
           mount           = true
           check           = true
           vol             = /home
           free            = false
           account         = true
   ```

6. Specify the data files to be used for printer accounting data by adding one separate line to each queue stanza in the `/etc/qconfig` file. The UNIX Performance component separates the print queue reports by using different print queue account file names.

   Example:

   ```
   dlp0:
           device = lp0
           acctfile = /var/adm/qacct.prquename <== your
   print queue name
   lp0:
           header = always
           trailer = never
           file = /dev/lp0
           backend = /usr/lpd/piobe
   ```

7. As the adm user, create the `/var/adm/acct/nite`, `/var/adm/acct/fiscal`, and `/var/adm/acct/sum` directories to collect daily and fiscal period records, by entering the following commands:

   ```
   su - adm
   cd /var/adm/acct
   mkdir nite fiscal sum
   exit
   ```

8. Make the daily accounting routines run automatically.

   The following statement, contained in /usr/lpp/epdm/etc/drlcron, shows you how to gather all log types. Add this statement to the cron file.

   ```
   0 2 * * * /usr/lpp/epdm/bin/drlgather acct perf >/dev/null 2>&1
   ```

   Update the root cron file using this command:

   ```
   crontab -e
   ```

   Include, also, in crontab file the **dodisk**, and **runacct** commands. For example:

   ```
   0 1 * * * /usr/sbin/acct/dodisk >/dev/null 2>&1
   30 1 * * * /usr/sbin/acct/runacct 2>/var/adm/acct/nite/accterr
   ```

   The first line starts disk accounting at 1:00 a.m. (0 1). The second line starts most accounting procedures and processes active data files at 1:30 a.m. (30 1) every day. You must have root user authority to edit the /var/spool/cron/crontabs/root file

9. Create a file named siteacct in the /var/adm directory, and insert these statements in it. If the file already exists and is a script, insert the statements in the existing file. The statements are:

   ```
   date=`date +"%m%d"`
   cat /var/adm/Spacct*.$date >/var/adm/acct/nite/opacct
   ```

   Set the file permissions and ownership:

   ```
   chmod 550 siteacct
   chown adm:adm siteacct
   ```

   If the /var/adm/siteacct file already exists and is a binary file (an executable file), use one of these methods instead:**Either:** Update /usr/sbin/acct/runacct to also call prsiteacct after siteacct, and create a file as described but name it prsiteacct. **or:** Rename the existing siteacct file to siteacct, and create file siteacct as described, and add after the two statements a call to siteacct as follows:

   ```
   /var/adm/siteacct
   ```

10. Start process accounting (if accounting is not already running on the AIX system). This will be done at start time as a result of the preceding change to the /etc/rc file. However, to start immediate process accounting, issue this command:

    ```
    nohup /usr/sbin/acct/runacct 2> /var/adm/acct/nite/accterr &
    ```

Setting up accounting for HP

About this task

Set up accounting on HP nodes by following the instructions in the appropriate manual. In addition, follow these steps:

Procedure

1. Create a file named runacct.local in the /usr/lib/acct directory, and insert the following statements in it. If the file already exists and is a script, insert the statements in the existing file. The statements to insert are:

   ```
   date=`date +"%m%d"`
   cat /usr/adm/Spacct*.$date >/usr/adm/acct/nite/opacct
   ```

   Set the file permissions and ownership:

   ```
   chmod 550 runacct.local
   chown adm:adm runacct.local
   ```

2. To run runacct.local, edit the file named /usr/lib/acct/runacct. If necessary, insert the following statement:

   ```
   [ -s /usr/lib/acct/runacct.local ] && /usr/lib/acct/runacct.local
   ```

3. Start process accounting (if accounting is not already running):

   ```
   /bin/su - adm -c /usr/lib/acct/startup
   ```

   You can also add this statement to the /etc/rc file to enable it to be executed at start time.

Setting up accounting for Sun Solaris

About this task

Set up accounting on Sun Solaris nodes by following the instructions in the appropriate manual. In addition, follow these steps:

Procedure

1. Ensure that during the operating system installation you installed the SUNWaccr and SUNWaccu accounting packages. If not, install them by running the following statements:

   ```
   pkginfo  -l  SUNWaccu
   pkginfo  -l  SUNWaccr
   ```

2. Create a file named runacct.local in the /usr/lib/acct directory, and insert the following statements in the file:

   ```
   date=`date +"%m%d"`
   cat  /var/adm/Spacct*.$date > /var/adm/acct/nite/opacct
   ```

   If the file already exists, insert these statements in the existing file.

3. Set the runacct.local file permissions and ownerships:

   ```
   chmod 550  runacct.local
   chown adm  runacct.local
   chgrp adm  runacct.local
   ```

4. To run runacct.local file, ensure that the following statement appears after the USEREXIT statement in the /usr/lib/acct/runacct file:

   ```
   [ -s /usr/lib/acct/runacct.local ] && /usr/lib/acct/runacct.local
   ```

   If the statement is missing, insert it after the USEREXIT statement.

5. Start process accounting (if accounting is not already running):

   ```
   /usr/bin/su - adm -c /usr/lib/acct/startup
   ```

   You can also add this statement to the /etc/rc file to enable it to be executed at start time.

Step 2: Create log files

About this task

By including drlgather commands in the cron file, you can schedule the creation of log files, by entering the following command:

```
/usr/lpp/epdm/bin/drlgather -c conf_day component_list
```

Where:

conf_day
: Is used with the configuration component, and specifies the day on which the configuration log file will be created. The creation of the log file occurs

on (default value) the first day of each calendar month. In the following example, the configuration log file will be created every fifth day:

```
-c"1 5 10 15 20 25"
```

component_list

Can be one or more of the following:

- acct (accounting subcomponent)
- conf (configuration subcomponent)
- error (error subcomponent)
- perf (performance subcomponent)

Procedure

1. For AIX, edit the root cron file using the command `crontab -e`. For HP, edit the root cron file using the following commands:

   ```
   cd /var/spool/cron/crontabs
   vi root
   crontab root  (to update the cron daemon)
   ```

   For Sun Solaris, edit the root cron file using the following commands:

   ```
   cd /usr/spool/cron/crontabs
   vi root
   crontab root  (to update the cron daemon)
   ```

   For the accounting, error, and performance subcomponents, the creation of the log files occurs at (default value) 02.00 hours each day.

   To verify that the updates were successful, enter:

   ```
   crontab -l
   ```

2. Select the subcomponents for which you wish to gather log files.

   The following example shows you how to automatically schedule log files for all subcomponents:

   ```
   0 2 * * * /usr/lpp/epdm/bin/drlgather -c1 acct error perf conf
   ```

   The next example shows you how to automatically schedule *only* the log files for the accounting and performance subcomponents:

   ```
   0 2 * * * /usr/lpp/epdm/bin/drlgather acct perf
   ```

   (This statement can also be found in `/usr/lpp/epdm/etc/drlcon`)

Step 3: Define the receiver of UNIX Performance component data

About this task

Procedure

1. Insert in the home directory .netrc file, the following entry:

   ```
   machine mvssystem login epdmuser password
   pwd
   ```

   Where the *mvssystem*, *epdmuser*, and *pwd* are the values for the OS/390 host user that will receive the data files. FTP uses this file when performing the file transfer, each night.

   Note: The value for password must be the same as the current password used on OS/390.

2. After editing the .netrc file, check that the file has private read/write permission only, by entering the command `chmod 600 .netrc`

3. Use the command `ping -c1 mvssystem`, to check that communication between the UNIX Performance component and OS/390 is possible. You can break from this command by typing Ctrl+C. If the `ping` command fails, you should add a host definition to the file `/etc/hosts`, as shown in the following example:

   ```
   9.99.99.99 mvssystem
   ```

   where 9.99.99.99 is the IP address, and *mvssystem* is the name of the OS/390 system.

Step 4: Customizing UNIX Performance component parameters

About this task

Edit the parameter file `/etc/drl.cfg` to change the default values supplied by the UNIX Performance component to those for your installation. If you are using the parameter file default values, you only need to change the value of `epdmHost`. All the subcomponents use these parameter file values.

Parameter file values for all subcomponents

```
logPath      = /var/epdm
exePath      = /usr/lpp/epdm/bin   (Do not change)
epdmHost     = yourmvs
epdmPrefix   = EPDMUNIX
minFree      = 2000
site         = bl=10204 lr=255 rec=vb cy pri=1 sec=5
maxRetries   = 72
retInterval  = 60
HostLabel    =
```

Parameter
Description

logPath
The name of directory where all the data log files will be stored.

exePath
The name of the directory where all commands are.

msgCatalog
The name of the file where all messages are stored.

epdmHost
Host name (TCP nodename or TCP internet address) of the OS/390 host to receive the UNIX Performance component data files.

epdmPrefix
Log data set prefix, which you set according to your own requirements.

minFree
Minimum free disk space in KB on /var/epdm before running the daily drlgather.

site OS/390 data set characteristics for the data files sent using TCP/IP FTP to OS/390.

maxRetries
When the UNIX Performance component data files are to be automatically transferred to OS/390 each night, the communication might be broken for some reason, or the OS/390 host might be temporarily unavailable. If that is the case, the routine `drlgather` will repeat attempt the file transfer of the data files for a number of times given by the parameter 'maxRetries'. Using

the default value for maxRetries, and with the retInterval set at its default value of 60 minutes, the file transfer retry period will, therefore, cover 3 days.

retInterval
Interval in seconds between ftp retries (default value = 30).

HostLabel
Set this parameter when there are more hosts that may have the same hostname (in an hacmp environment). If the HostLabel parameter has not any value, the product runs the 'uname -n' command, otherwise it returns the HostLabel value.

Note:

1. Do not change the first two parameters `logPath`, and `exPath` as they are for internal use only.
2. You must specify the node name in the `epdmHost` parameter and the data set prefix in the `epdmPrefix` parameter. This is the name of the host and user to receive the data via TCP/IP FTP. These names must be the same as the values of node name and user ID in the .netrc file.

Parameter file values for Performance component

The parameter file values used by the performance component (only), are as follows:

```
perfInterval = 60
perfCount    = 10
diskInterval = 60
```

Parameter
Description

perfInterval
The program drlperfd issues `vmstat` and `iostat` commands, which contain the perfInterval value as the interval parameter. The default interval between samples is 60 seconds.

perfCount
The program drlperfd samples vmstat and iostat data, at the interval values shown. This data is saved in the file `/var/epdm/`*`node_name`*`.perf.d`*`yymmdd.n`*, as follows:

- The minimum, maximum and average values during the perfCount interval
- The number of users and processes
- The usage of paging space

The default interval between samples is 10 minutes.

diskInterval
The program drlperfd issues the commands `df` and `lsvg`, and save file space and volume group usage in the file `/var/epdm/`*`node_name`*`.disk.d`*`yymmdd. n`*. The default interval between samples is 60 minutes.

Table 19 on page 98 shows the directory created by the `drlcrfs` command explained in item 7 of "Step 2: Transfer Tivoli Decision Support for z/OS feature code to UNIX" on page 88.

Table 19. Creating a directory from the drlcrfs command

Directory	Description	Files to be created when UNIX Performance component running
/var/epdm/	This directory is created by the command `drlcrfs`. The directory resides in a new separate file system, generated and mounted over `/var/epdm` by an installation procedure. No files are loaded in this directory at installation. The directory will be used to hold work files, data files and log files. The data files will be deleted as soon as they are successfully transmitted.	**work files:** (see Note 1) drlerrpt.time (see Note 1) *node_name*.perf.work *node_name*.disk.work (see Note 1) *node_name*.perfio.work *node_name*.perfvm.work .sk **log data files:** (see Notes 2, 3, 4) *node_name*.pacct.d*yymmdd*.1 *node_name*.wtmp.d*yymmdd*.1 *node_name*.dtmp.d*yymmdd*.1 *node_name*.qacct.d*yymmdd*.1 (see Note 1) *node_name*.errpt.d*yymmdd*.1 (see Note 1) *node_name*.perf.d*yymmdd*.1 *node_name*.disk.d*yymmdd*.1 (see Note 1) *node_name*.conf.d*yymmdd*.1 (see Note 1) *node_name*.epdmunix.d*yymmdd*.1 (see Note 6) .sk **message log files:** (see Notes 2, 3) *node_name*.drlperfd.errorlog *node_name*.log.d*yymmdd* *node_name*.errorlog.d*yymmdd* .sk **ftp work files:** (see Notes 2, 3, 5) *node_name*.ftpwork.d*yymmdd* *node_name*.ftpmsg.d*yymmdd* *node_name*.ftperr.d*yymmdd*

Notes:

1. AIX only
2. *node_name* is the name of the node.
3. *yymmdd* is the current date.
4. These files will be removed after they have been transferred to OS/390.
5. These files will be removed after the log data files have been transferred to OS/390. If the transfer to OS/390 is not successful, these files contain details of the failure.
6. This file contains the 8 files listed before it (*node_name*.pacct.d*yymmdd*.1, *node_name*.wtmp.d*yymmdd*.1, and so on) and is the file to be transmitted.

Step 5: Start the drlperfd daemon

About this task

The processes that will collect performance and disk space data during the day, are initiated by program `drlperfd`.

You should now:

Procedure

1. Start the 4 background processes that gather performance and disk space data, by entering the command `drlactperfd`.

   If you enter the drlactperfd command without parameters on an AIX platform, the perf and disk data will be collected. You can collect them also running the "drlactperfd perf disk" command. If you want to collect only the disk or the perf data, enter the command `drlactperfd disk` or `drlactperfd perf`.

To start the trace, start the `drlactperfd` with the "trace" option as follows: `drlactperfd perf disk trace`.

2. Enter the command `drlpsperfd` to display the status of the UNIX Performance component processes. The four processes that should be listed, are shown in "drlpsperfd" on page 102

Results

Performance data will now be entered in the following files:

- work files
  - *node_name*.perf.work
  - *node_name*.disk.work
  - *node_name*.perfio.work
  - *node_name*.perfvm.work
- log data files
  - *node_name*.perf.d*yymmdd*
  - *node_name*.disk.d*yymmdd*

The file *node_name*.drlperfd.errorlog contains possible error messages. This file remains empty providing the `drlactperfd` command runs successfully.

The file *node_name*.drlperfd.log contains information messages which inform you, for example, of the files that have been transferred to OS/390.

Step 6: Invoke drlperfd at start time

About this task

(**performance component only**)

To automatically start the drlperfd program at IPL (boot) time, perform these steps:

Procedure

1. For AIX, enter the command `mkitab epdm:2:wait:/etc/rc.epdmunix`. For HP and Sun operating systems, edit /etc/inittab and add the epdm line at the end as shown.

```
#       1.28  com/cfg/etc/inittab, bos, bos320 10/3/00 10:46:51
#   COMPONENT_NAME: CFGETC
#
#   ORIGINS: 3, 27
.
.
.
epdm:2:wait:/etc/rc.epdmunix
```

2. Copy rc.epdmunix to /etc/ using the command `cp /usr/lpp/epdm/etc/ rc.epdmunix /etc/`.

Description of daily tasks performed by UNIX Performance component

During the day, UNIX Performance component has four processes running which are started by one program. UNIX is set up so that these four processes are autostarted at each IPL (boot-time), as described in "Step 6: Invoke drlperfd at start time." The program that starts the four processes is `drlperfd`, which periodically logs:

- Various performance data to the daily file /var/epdm/*node_name*.perf.d*yymmdd*.*n* (the default value is every 10 minutes).
- Disk space data to the daily file /var/epdm/*node_name*.disk.d*yymmdd*.*n* (the default is every 60 minutes).

Note: An explanation of *node_name*, *yymmdd*, and *n* is given in Table 19 on page 98..

Each night, UNIX accounting routines and the UNIX Performance component routines are run. These routines are started from the UNIX cron daemon at default times (which can be customized by you), in the root's cron file.

Time	Action
01:00	UNIX starts the dodisk command via cron.
01:30	UNIX starts the runacct command via cron.
02:00	UNIX starts the drlgather command via cron. Command drlgather calls the following routines: • drlpacct, which reads yesterday's process accounting file and produces an ASCII file suitable for sending to OS/390. • drlwtmp, which reads yesterday's login/logout accounting file and produces an ASCII file suitable for sending to OS/390. • drldtmp, which reads yesterday's disk accounting file and produces an ASCII file suitable for sending to OS/390. • drlqacct, which reads yesterday's print queue accounting files and produces an ASCII file suitable for sending to OS/390. • drlerrpt, which prints the AIX error log since previous print and produces an ASCII file suitable for sending to OS/390. • drlconf, which produces a hardware and software configuration file in ASCII format for sending to OS/390. This is performed once per month. • drlsend, which transmits a log file containing the concatenated ASCII files shown, together with the performance and disk space files created by drlperfd, to OS/390 (using TCP/IP ftp). If the link to OS/390 is temporarily down, or if ftp receives a time out during the connect from UNIX to OS/390, the file will be transmitted (by default) one hour later and this will be repeated (by default) 72 times.
02:30	UNIX starts the monacct command via cron daemon.

Following is an example of AIX crontab schedule:

```
0 1 * * * /usr/sbin/acct/dodisk >/dev/null 2>&1
30 1 * * * /usr/sbin/acct/runacct >/dev/null 2>/var/adm/acct/nite/accterr
0 2 * * * /usr/lpp/epdm/bin/drlgather -c1 acct error perf conf >/dev/null 2>&1
30 2 * * * /usr/sbin/acct/monacct >/dev/null 2>&1
```

Operating the UNIX Performance component on UNIX

This section describes:

- The UNIX Performance component logs
- The commands that you use to operate the UNIX Performance component on UNIX (see "AIX commands to use if UNIX Performance component installed" on page 102).

Description of UNIX Performance component files

Examine the /var/epdm directory to see how the nightly file transfer routines have performed. Look for the files *node_name*.log.d*yymmdd*, where *node_name* is the name of the node and *yymmdd* is current date. There should be one information log file per day.

By looking into a particular information log file you can see:

- What files were sent that day.
- How big they were.
- If and why the ftp file transfer was not completed.

This is an example of the /var/epdm directory:

```
ls -l /var/epdm
total 1544
-rw-r--r--   1 root     system       770 Nov 24 19:12 drl.cfg
-rw-r--r--   1 root     system      7847 Feb 07 10:43 node1.disk.work
-rw-r--r--   1 root     system         0 Jan 02 19:25 node1.drlperfd.errorlog
-rw-r--r--   1 root     system    618581 Jan 22 02:11 node1.epdmunix.d950127.1
-rw-r--r--   1 root     system        93 Jan 22 02:12 node1.ftpwork.d950127
-rw-r--r--   1 root     system       436 Jan 22 02:12 node1.ftpmsg.d950127
-rw-r--r--   1 root     system       224 Jan 22 02:12 node1.ftperr.d950127
-rw-r--r--   1 root     system         0 Jan 22 02:00 node1.errorlog.d950122
-rw-r--r--   1 root     system         0 Jan 23 02:00 node1.errorlog.d950123
-rw-r--r--   1 root     system         0 Jan 24 02:00 node1.errorlog.d950124
-rw-r--r--   1 root     system         0 Jan 25 02:00 node1.errorlog.d950125
-rw-r--r--   1 root     system         0 Jan 26 02:00 node1.errorlog.d950126
-rw-r--r--   1 root     system         0 Jan 27 02:00 node1.errorlog.d950127
-rw-r--r--   1 root     system         0 Jan 28 02:00 node1.errorlog.d950128
-rw-r--r--   1 root     system         0 Jan 29 02:00 node1.errorlog.d950129
-rw-r--r--   1 root     system         0 Jan 22 02:00 node1.errorlog.d950122
-rw-r--r--   1 root     system       168 Feb 01 02:01 node1.errorlog.d950201
-rw-r--r--   1 root     system         0 Feb 02 02:00 node1.errorlog.d950202
-rw-r--r--   1 root     system         0 Feb 03 02:00 node1.errorlog.d950203
-rw-r--r--   1 root     system        11 Feb 07 19:10 node1.errpt.time
-rw-r--r--   1 root     system      2307 Jan 22 02:02 node1.log.d950122
-rw-r--r--   1 root     system      2321 Jan 23 02:02 node1.log.d950123
-rw-r--r--   1 root     system      2302 Jan 24 02:03 node1.log.d950124
-rw-r--r--   1 root     system      2409 Jan 25 02:04 node1.log.d950125
-rw-r--r--   1 root     system      2043 Jan 26 02:02 node1.log.d950126
-rw-r--r--   1 root     system      2312 Jan 27 02:03 node1.log.d950127
-rw-r--r--   1 root     system      2351 Jan 28 02:03 node1.log.d950128
-rw-r--r--   1 root     system      2234 Jan 29 02:02 node1.log.d950129
-rw-r--r--   1 root     system      2287 Jan 30 02:04 node1.log.d950130
-rw-r--r--   1 root     system      2357 Jan 31 02:03 node1.log.d950131
-rw-r--r--   1 root     system      2308 Feb 01 02:02 node1.log.d950201
-rw-r--r--   1 root     system      2319 Feb 02 02:04 node1.log.d950202
-rw-r--r--   1 root     system      2401 Feb 03 02:02 node1.log.d950203
-rw-r--r--   1 root     system     35293 Feb 07 11:08 node1.perf.work
-rw-r--r--   1 root     system       111 Feb 07 11:07 node1.perfio.work
-rw-r--r--   1 root     system       116 Feb 07 11:07 node1.perfvm.work
```

In this example there is a communication problem on January 27, so the files for this day have not been sent. You can check the contents of the files node1.ftpwork.d950127, node1.ftpmsg.d950127, and node1.ftperr.d950127 for details of the problem. The merged log data file for this date has been retained by the system. When the problem has been solved, you can then manually send the log data file (node1.epdmunix.d950127.1), using the command `drlsend`.

You can also see that an error has occurred on February 01, because the file node1.errorlog.d950201 is not empty. You can check the contents of this file for details of the problem.

Obsolete log files must be deleted manually.

AIX commands to use if UNIX Performance component installed

Use the following four commands when operating the UNIX Performance component on UNIX.

drlpsperfd

This command displays the status of the UNIX Performance component processes in UNIX:

```
drlpsperfd
The following drlperfd processes are running
    USER COMMAND       PID       PPID
    root drlperfd      10200        1
    root drlperfd      11735    10200  (Not for HP or Sun)
    root drlvmstat     17153    11735
    root drliostat     19970    11735
```

drlactperfd

Use this command to start the performance component processes in UNIX. It starts the drlperfd daemon. This daemon will create performance log files. Command drlactperfd can also be used to restart the drlperfd daemon.

drldeactperfd

Use this command to stop the performance component processes in UNIX.

drlsend

If there is a communication problem, the UNIX Performance component routines automatically attempt to re-send the files. However, for the following reasons you might need to *manually* re-send the data files to OS/390:

- If AIX has been rebooted and was "down" at the time of a retry event, the event will never be rescheduled.
- If the ftp retry period has been exceeded. The ftp retry period is defined using the parameters maxRetries and retInterval in the parameter file `/etc/drl.cfg`.

All data files for those days in which the data files were not sent, remain on the `/var/epdm` directory. After the communication problems has been corrected, the data files for a particular day can be sent using the `drlsend` command. The `drlsend` command deletes the files after they have been successfully transferred. Issue this command with the `date` parameter, as follows:

```
drlsend  yymmdd
```

where *yymmdd* can be found in the corresponding file names, in the `/var/epdm` directory.

Installing UNIX Performance component on the z/OS system

About this task

When you install UNIX Performance component subcomponents, Tivoli Decision Support for z/OS will install the required log and record definitions, record procedures, and update definitions to the product system tables. Tivoli Decision Support for z/OS will also install the predefined tables (described in Chapter 12, "Data, lookup, and control tables," on page 133) and reports (described in Chapter 14, "Reports," on page 151). To install the UNIX subcomponents, use the Administration dialog.

Perform steps 1 to 4, as follows:

Procedure

1. From the Tivoli Decision Support for z/OS Administration window (Figure 32), select 2, Components and press Enter.

```
  Other  Utilities  Help
-------------------------------------------------------------------------------
                  Tivoli Decision Support for OS/390 Administration

Select one of the following.  Then press Enter.


              2_  1.  System
                  2.  Components
                  3.  Logs
                  4.  Tables
                  5.  Reports









Command ===> _____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F9=Swap     F10=Actions  F12=Cancel
```

Figure 32. Tivoli Decision Support for z/OS Administration window

The Components window is displayed, (as shown in Figure 33).

```
  Components  Other  Help
-------------------------------------------------------------------------------
                                  Components                ROW 1 TO  13 OF 47

Select one or more components. Then press Enter to Open component.

/   Components                                              Status     Date
_   Network Line Utilization Component                      Installed  00-03-19
_   Network NetView FTP Component                           Installed  00-03-19
_   Network NPM Internal Utilization Component              Installed  00-03-19
_   Network NV/SM Internal Utilization Component            Installed  00-03-19
_   Network Problem Component                               Installed  00-03-19
_   Network RTM Response Time Component                     Installed  00-03-19
_   Network Service Component                               Installed  00-03-19
_   DB2 Component                                           Installed  00-03-17
_   Information/Management Component                        Installed  00-03-17
/   UNIX Accounting Component
_   UNIX Configuration Component
_   UNIX Error Component
_   UNIX Performance Component

Command ===> _____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F5=New       F6=Install   F7=Bkwd
 F8=Fwd       F9=Swap     F10=Actions  F12=Cancel
```

Figure 33. Components window

2. From the Components window, select the components to install (here, the AIX accounting component) and press F6.
3. The Installation Options window is displayed, (as shown in Figure 34 on page 104).

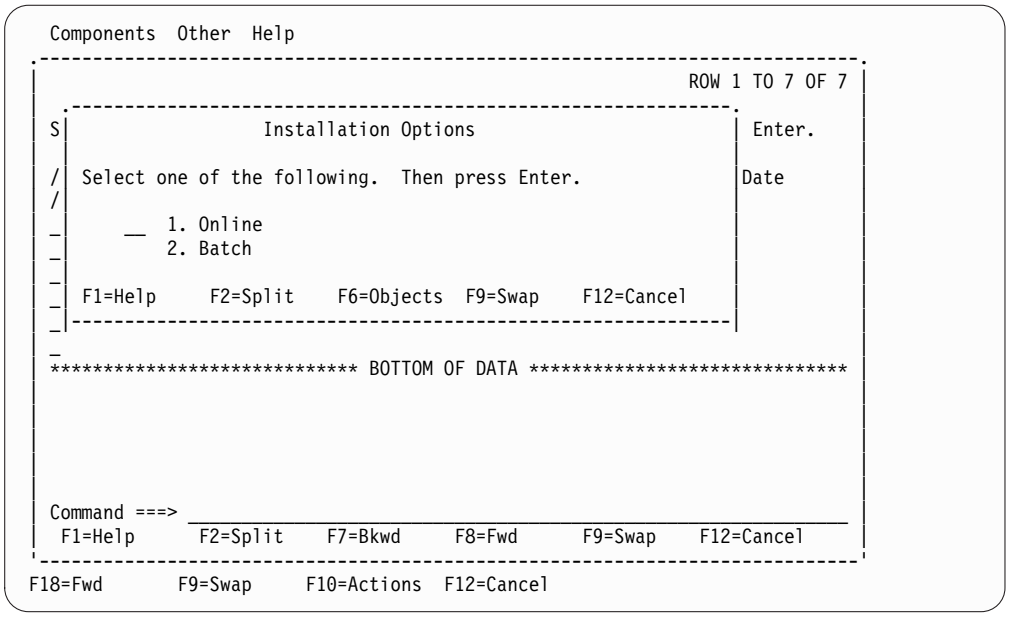


Figure 34. Installation Options window

4. Using the component-installation procedure in the *Administration Guide and Reference*, SH19-6816 specify whether the component is to be installed online, or in batch mode.

   Batch mode installation results in less output than in online. Furthermore, for online installation your terminal will be blocked for the duration of the installation. Therefore, it is recommended that you install components in batch.

Updating the lookup tables

About this task

All the subcomponents of the UNIX Performance component include lookup tables that you can customize to specify the groupings you want reflected in your reports.

If you specify online installation, Tivoli Decision Support for z/OS displays the Lookup Tables window. To edit a lookup table using ISPF edit, select a table and press Enter.

If you specify batch mode installation, you can edit the lookup tables using the ISPF editor, *after* the component is installed. To enter an ISPF edit from Tivoli Decision Support for z/OS, you should:

Procedure

1. Select 'Tivoli Decision Support for z/OS Administration' from the Tivoli Decision Support for z/OS Primary Menu
2. Select '4. Tables'
3. Select the lookup table that you wish to edit, select the 'Edit' pull-down, and press Enter
4. Select '3. ISPF Editor' from the Edit pull-down.

Results

With the UNIX Performance component, customization involves updating the XACCT_CONNECT_TYPE, X_NODE_NAME, and XACCT_PRINT_TYPE lookup tables, described in the following sections.

Updating XACCT_CONNECT_TYPE

This lookup table has information about connect types. It is used during reporting, to translate hexadecimal connect codes to a descriptive text.

A sample of the lookup table contents is given in "XACCT_CONNECT_TYPE" on page 144.

Updating X_NODE_NAME

This lookup table contains UNIX node names, accounting groups, Internet addresses, and period plans. It is used during the collect, to translate the node name to its accounting group and period plan.

If you are installing online, you will be given the opportunity to edit the contents using the ISPF editor. A sample of the lookup table contents is given in "X_NODE_NAME" on page 145. Add the host names of all your UNIX nodes for which you are going to do reporting.

Note: If you have many UNIX nodes, you can obtain a listing of the node names, by entering the following command on one of your AIX nodes:

```
hostent -S
```

Updating XACCT_PRINT_TYPE

Node names and print queue names are grouped together into print types. This lookup table contains each print type.

A sample of the lookup table contents is given in "XACCT_PRINT_TYPE" on page 145.

After installation is completed, Tivoli Decision Support for z/OS returns you to the Components window, and the Status field indicates that the component is installed.

Collecting data under Tivoli Decision Support for z/OS

About this task

There are various methods you can define yourself, to set up the JCL for the COLLECT procedure. This section, however, describes a method that uses a GDG (generation data set). It explains:

- How a Generation Data Group (GDG) is created.
- How the files received from UNIX are renamed, so they can be used with a GDG.
- The JCL for a general *collect* job, which uses a GDG to collect data into Tivoli Decision Support for z/OS tables.
- The JCL for a *collect* job with specified Tivoli Decision Support for z/OS tables to be updated.

Creating your Generation Data Groups

About this task

You are recommended to use a generation data group (GDG) to avoid the large amount of JCL renaming work that would otherwise be required each time a collect was run. This is especially true if you work with a system containing many nodes.

The GDG must be created *before you run the first collect*. Thereafter, the *same* GDG can be used with each collect. The following job is provided to create the GDG:

`DRL180.SDRLCNTL(DRLJXGDG)`

Renaming files sets sent from UNIX for use with a GDG

About this task

In a large UNIX network, there will be many files sent from UNIX nodes to the host OS/390 each night. Each UNIX file transferred to OS/390 is given a unique data set name according to this naming convention:

prefix`.`*node_name*`.EPDMUNIX.D`*yymmdd*

Where:

prefix The high-level qualifier (for example, the user ID), for receiving files from UNIX.

node_name
First qualifier from the UNIX node name.

yymmdd
Creation date of the file in UNIX.

This naming convention ensures that no data is lost by overlaying existing data sets, and makes it easy to identify the origin of the data.

Before *each* collect of data, you must rename the files sent from UNIX so they become a generation data set within a GDG. The GDG is then the input to the collect job. The following job is provided to do this:

`DRL182.SDRLCNTL(DRLJXPCO)`

This job uses the REXX program DRLEXRNM, which renames each UNIX file to a generation data set within the GDG. The GDG has the following naming convention:

prefix`.COLLECT.EPDMUNIX`

Note: You should customize this job to meet your own naming conventions. Commented within the job DRL182.SDRLCNTL(DRLJXPCO) are detailed instructions on how to do this.

Example of renaming UNIX files

This example shows how files from the nodes `ADAM`, `BAKER`, and `CESAR`, are renamed to generation data sets, within the GDG with the name `USER1.COLLECT.EPDMUNIX`.

```
GDG USER1.COLLECT.EPDMUNIX empty
Following datasets found for USER1.*.EPDMUNIX.D* and renamed to
USER1.COLLECT.EPDMAIX

USER1.ADAM.EPDMUNIX.D921214           G0001V00
USER1.ADAM.EPDMUNIX.D921215           G0002V00
```

```
USER1.BAKER.EPDMUNIX.D921214          G0003V00
USER1.CESAR.EPDMUNIX.D921213          G0004V00
USER1.CESAR.EPDMUNIX.D921214          G0005V00
USER1.CESAR.EPDMUNIX.D921215          G0006V00

Number of datasets:6
```

Note:

1. The program DRLEXRNM used the search criterion 'USER1.*.EPDMUNIX.D*'.
2. The generation numbers are given in the right column (G0001V00, G0002V00, and so on).
3. Generation data sets are given the next sequential generation number (existing generations are retained). For example, if the generation data set with generation number G0006V00 currently exists in the GDG, new data sets will be given the generation numbers G0007V000, G0008V000, and so on.

General collect job

The member `DRL180.SDRLCNTL(DRLJCOLL)` contains sample JCL for a general collect.

Another way to obtain the COLLECT JCL, is to:

1. Select 'Tivoli Decision Support for z/OS Administration' from the Tivoli Decision Support for z/OS Primary Menu.
2. Select '3. LOGS'.
3. Select 'UNIX' from the list of logs, select the 'Utilities' pull-down, and press Enter.
4. Select '1. Collect'.
5. Type your installation-specific information in this window, and press Enter.

The JCL given in Figure 35 on page 108 is the general collect job which uses (in the DRLLOG statement) the example of a GDG for the nodes `ADAM`, `BAKER`, and `CESAR`.

```
//JOBCARD
//*                                                              *
//*  Notes:                                                      *
//*    Before you submit the job:                                *
//*    - Check the Tivoli Decision Support for z/OS and DB2   *
//*      data set names.                                         *
//*    - Check the DB2 subsystem name (default is DSN)           *
//*      and Tivoli Decision Support for z/OS                 *
//*      system table prefix (default is DRLSYS).                *
//*    - Insert the correct collect statement in DRLIN           *
//*      (as described above).                                   *
//*    - Specify the name of the log data set (or GDG) in        *
//*      DRLLOG. In this example, all existing generations       *
//*      are to be collected for files from UNIX                 *
//*                                                              *
//****************************************************************
//COLLECT  EXEC PGM=DRLPLC,PARM=('SYSTEM=DB2A SYSPREFIX=DRLSYS')
//STEPLIB  DD DISP=SHR,DSN=DRL180.SDRLLOAD
//         DD DISP=SHR,DSN=db2loadlibrary
//DRLIN    DD *

COLLECT UNIX;

//DRLLOG   DD DISP=SHR,DSN=USER1.COLLECT.EPDMUNIX
//DRLOUT   DD SYSOUT=*,DCB=(RECFM=F,LRECL=80)
//DRLDUMP  DD SYSOUT=*,DCB=(RECFM=F,LRECL=80)
/*
```

Figure 35. General COLLECT job

Collect job with specified Tivoli Decision Support for z/OS tables

The *Administration Guide and Reference* describes how to optimize Tivoli Decision Support for z/OS data collection by specifying only the needed tables in the INCLUDE (or EXCLUDE) clauses of the COLLECT statement.

The example in Figure 36 on page 109 shows how to selectively include two of the Resource Accounting data tables.

```
//JOBCARD
//*                                                              *
//*  Note:                                                       *
//*    - instead of entering each of the                         *
//*      2 table names (below), you can use:                     *
//*        INCLUDE LIKE 'DRL.XACCT_D%'                           *
//*                                                              *
//**************************************************************
//COLLECT  EXEC PGM=DRLPLC,PARM=('SYSTEM=DB2A SYSPREFIX=DRLSYS')
//STEPLIB  DD DISP=SHR,DSN=DRL180.SDRLLOAD
//         DD DISP=SHR,DSN=db2loadlibrary
//DRLIN    DD *

COLLECT UNIX
        INCLUDE        DRL.XACCT_DISK_D,
                       DRL.XACCT_DISK_M,
        ;

//DRLLOG   DD DISP=SHR,DSN=USER1.COLLECT.EPDMUNIX
//DRLOUT   DD SYSOUT=*,DCB=(RECFM=F,LRECL=80)
//DRLDUMP  DD SYSOUT=*,DCB=(RECFM=F,LRECL=80)
/*
```

Figure 36. Sample COLLECT job with an INCLUDE clause

Backup and delete of generation data sets

After a successful collect, you should back up and delete all generation data sets contained in the GDG. There are 2 reasons why you should do this:

1. It will enable you to use the same procedure without any JCL changes, when running the next collect job.
2. Unless you delete the existing generation data sets, the next collect job will reuse old generation data sets.

Testing the installation

About this task

Before starting the daily use of the UNIX Performance component, run a few tests to check that:

Procedure

1. The installation was successful.
   - Tivoli Decision Support for z/OS is collecting the correct data
   - The data is being stored correctly
   - The correct data is being used for the creation of reports
2. The lookup tables contain appropriate values.

Results

Refer to the *Administration Guide and Reference*, SH19-6816 regarding the steps involved in testing component installation.

Putting the feature into production

After you run the tests and verify that the installation is successful, you can put the UNIX Performance component and its subcomponents into production.

Figure 37 shows the daily steps involved in using Tivoli Decision Support for z/OS.

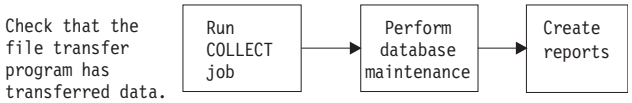


Figure 37. Daily steps involved in using Tivoli Decision Support for z/OS

You can run reports in batch, after setting batch parameters for each report using the administration dialog.

For detailed information about these steps, refer to the *Administration Guide and Reference and Reference*.

Considering which Linux subcomponents to install

The Linux Performance component is divided into three subcomponents:
- Configuration (AIX only)
- Error (AIX only)
- Performance

Installing the Linux Performance component on your Linux nodes

This section gives you practical information on how to install Tivoli Decision Support for z/OS on your Linux nodes.

Note that the steps are very close to the ones of the Unix nodes. For more detailed information, please refer to them where applicable.

Step 1: Check Linux requirements

This section lists the Linux software requirements.

Software requirements
About this task

The Linux Performance component requires one of the following programs:
- Linux REDHAT 9 or earlier
- Linux SUSE 10.1 or earlier
- TURBOLinux 10 (Kernel 2.2.18) or earlier

Step 2: Transfer Tivoli Decision Support for z/OS feature code to Linux

About this task

The Linux part of the Linux Performance component is distributed in the DRL180, SDRLWS (DRLLINUX) for Linux SMP target library member.

Download these `tar` files using these steps:

Procedure

1. Log in as root user on a Linux node.
2. Enter the commands:

```
mkdir /usr/lpp/epdm
cd /usr/lpp/epdm
```

3. Start an `ftp` session to your OS/390 host, and receive the file from the OS/390 user where you installed Tivoli Decision Support for z/OS

   Notes:

   a. You need to specify the IP address/name of the OS/390 system where Tivoli Decision Support for z/OS was installed and where the tar file is located.

   b. You need to specify the TSO userid that has read access to the tar file.

4. When the Linux Performance component code has been received by one Linux node, send the tar files to all other Linux nodes where the Linux Performance component is to be installed.
5. On each of the Linux nodes to which the tar files have been sent, enter the following tar command to unpack the epdm*LINUX* tar files and create the files in the directories `tar -xvf epdmLINUX.tar`
6. (Optional) After completing the previous step you can remove the tar files using the command `rm epdmLINUX.tar`. (If you need the tar files in the future, they still exist on OS/390 and can be used again).
7. Create file system or directory for log files.
   - The command `drlcrfs` is provided to create a file system with the name /var/epdm/, in volume group rootvg. The size of the file system is determined by an argument used with the `drlcrfs`. The Linux Performance component uses the file system /var/epdm/ to store work and data files until they have been sent to OS/390.

     The `drlcrfs` command is also used to mount the file system using /var/epdm as the mount point.

     By having the /var/epdm/ directory in a separate file system, there is no risk of obtaining a "file system full" condition on file system /var (which would stop Linux processing).

     For example, to create and mount a new file system with size 50MB, enter:

     ```
     /usr/lpp/epdm/bin/drlcrfs 50
     ```

     Confirm that the new file system has been created, by entering the following command:

     ```
     df /var/epdm
     ```

8. The drl.cfg parameter file is distributed as /usr/lpp/epdm/etc/drl.cfg. Copy the parameter file to /etc/drl.cfg by entering `cp /usr/lpp/epdm/etc/drl.cfg /etc/drl.cfg`.

Capturing Linux Performance component data and transmitting to OS/390

About this task

This section describes:

- The steps you must perform to set up the Linux Performance component system to capture performance data, and to transmit the data to the OS/390 system:
  Step 1: Create log files
  Step 3: Define the receiver of Linux Performance component data
  Step 4: Customize Linux Performance component parameters
  Step 5: Start the drlperfd daemon (performance subcomponent)

Note: Where a step is for a specific subcomponent only, the subcomponent name is given in parentheses in the heading for the step. Otherwise, you must carry out the step for all the subcomponents (configuration, error, and performance).

Step 1: Create log files

About this task

By including `drlgather` commands in the cron file, you can schedule the creation of log files, by entering the following command:

```
/usr/lpp/epdm/bin/drlgather -c conf_day component_list
```

where:

conf_day
is used with the configuration component, and specifies the day on which the configuration log file will be created. The creation of the log file occurs on (default value) the first day of each calendar month. In the following example, the configuration log file will be created every fifth day:

```
-c"1 5 10 15 20 25"
```

component_list
can be one or more of the following:

- conf (configuration subcomponent)
- error (error subcomponent)
- perf (performance subcomponent)

Procedure

1. For AIX, edit the root cron file using the command `crontab -e`.

   For the accounting, error, and performance subcomponents, the creation of the log files occurs at (default value) 02.00 hours each day.

   To verify that the updates were successful, enter `crontab -l`.
2. Select the subcomponents for which you wish to gather log files.

   The following example shows you how to automatically schedule log files for all subcomponents:

   ```
   0 2 * * * /usr/lpp/epdm/bin/drlgather -c1 error perf conf
   ```

   The next example shows you how to automatically schedule *only* the log files for the accounting and performance subcomponents:

   ```
   0 2 * * * /usr/lpp/epdm/bin/drlgather perf
   ```

   (This statement can also be found in /usr/lpp/epdm/etc/drlcon)

Step 2: Define the receiver of Linux Performance component data

About this task

Procedure

1. Insert in the home directory .netrc file, the following entries:

   ```
   machine mvssystem login epdmuser password
   secret
   ```

2. Change the *mvssystem*, *epdmuser*, and *secret* to the values corresponding to the OS/390 host user that will receive the data files. Ftp uses this file when performing the file transfer, each night.

   Note: The value for password *must* be the same as the current password used on OS/390
3. After editing the .netrc file, check that the file has private read/write permission only, by entering the command `chmod 600 .netrc`.
4. Use the command `ping -c1 mvssystem`, to check that communication between the Linux Performance component and OS/390 is possible. You can break from this command by typing Ctrl+C. If the `ping` command fails, you should add a host definition to the file /etc/hosts, as shown in the following example:

   ```
   9.99.99.99 mvssystem
   ```

   where '9.99.99.99' is the IP address, and 'mvssystem' is the name of the systemOS/390.

Step 3: Customizing Linux Performance component parameters

About this task

Edit the parameter file /etc/drl.cfg to change the default values supplied by the Linux Performance component to those for your installation. If you are using the parameter file *default* values, you only need to change the value of `epdmHost`. All the subcomponents use these parameter file values.

Parameter file values for all subcomponents

```
logPath      = /var/epdm
exePath      = /usr/lpp/epdm/bin   (Do not change)
epdmHost     = yourmvs
epdmPrefix   = EPDMLINUX
minFree      = 2000
site         = bl=10204 lr=255 rec=vb cy pri=1 sec=5
maxRetries   = 72
retInterval  = 60
```

Parameter
Description

logPath
The name of directory where all the data log files will be stored.

exePath
The name of the directory where all commands are.

msgCatalog
The name of the file where all messages are stored.

epdmHost
Host name (TCP nodename or TCP internet address) of the OS/390 host to receive the Linux Performance component data files.

epdmPrefix
Log data set prefix, which you set according to your own requirements.

minFree
Minimum free disk space in KB on /var/epdm before running the daily drlgather.

site OS/390 data set characteristics for the data files sent using TCP/IP FTP to OS/390.

maxRetries
When the Linux Performance component data files are to be automatically transferred to OS/390 each night, the communication might be broken for some reason, or the OS/390 host might be temporarily unavailable. If that is the case, the routine `drlgather` will reattempt the file transfer of the data files for a number of times given by the parameter 'maxRetries'. Using the default value for maxRetries, and with the retInterval set at its default of 60 minutes, the file transfer retry period will, therefore, cover 3 days.

retInterval
Interval in seconds between ftp retries (default value = 30).

Note:

1. Do not change the first two parameters `logPath`, and `exPath` as they are for internal use only.
2. You must specify the node name in the `epdmHost` parameter and the data set prefix in the `epdmPrefix` parameter. This is the name of the host and user to receive the data via TCP/IP FTP. These names must be the same as the values of node name and user ID in the .netrc file.

Parameter file values for performance component

The parameter file values used by the performance component (only), are as follows:

```
perfInterval = 60
perfCount    = 10
diskInterval = 60
```

Parameter
Description

perfInterval
The program drlperfd issues `vmstat` and `iostat` commands, which contain the perfInterval value as the interval parameter. The default interval between samples is 60 seconds.

perfCount
The program drlperfd samples vmstat and iostat data, at the interval values shown. This data is saved in the file `/var/epdm/`*`node_name`*`.perf.d`*`yymmdd.n`*, as follows:

- The minimum, maximum and average values during the perfCount interval
- The number of users and processes
- The usage of paging space

The default interval between samples is 10 minutes.

diskInterval
The program drlperfd issues the commands `df` and `lsvg`, and save file space and volume group usage in the file `/var/epdm/`*`node_name`*`.disk.d`*`yymmdd. n`*. The default interval between samples is 60 minutes.

Step 4: Start the drlperfd daemon

About this task

(**performance component only**)

The processes that will collect performance and disk space data during the day, are initiated by program `drlperfd`.

You should now:

Procedure

1. Start the 4 background processes that gather performance and disk space data, by entering the command `drlactperfd`.
2. Enter the command `drlpsperfd` to display the status of the Linux Performance component processes. The four processes that should be listed, are shown in "drlpsperfd" on page 102.

Results

Performance data will now be entered in the following files:

- work files
  - *node_name*.perf.work
  - *node_name*.disk.work
- log data files
  - *node_name*.perf.d*yymmdd*
  - *node_name*.disk.d*yymmdd*

The file *node_name*`.drlperfd.errorlog` contains possible error messages. This file remains empty providing the `drlactperfd` command runs successfully.

The file *node_name*`.drlperfd.log` contains information messages which inform you, for example, of the files that have been transferred to OS/390.

Commands to use if Linux Performance component installed

Use the following four commands when operating the Linux Performance component on Linux.

drlpsperfd

This command displays the status of the Linux Performance component processes in Linux.

```
drlpsperfd
The following drlperfd processes are running
    USER COMMAND       PID       PPID
    root drlperfd      10200        1
    root drlperfd      11735    10200
    root drlvmstat     17153    11735
    root drliostat     19970    11735
```

drlactperfd

Use this command to start the performance component processes in Linux. It starts the drlperfd daemon. This daemon will create performance log files. Command drlactperfd can also be used to restart the drlperfd daemon.

drldeactperfd

Use this command to stop the performance component processes in Linux.

drlsend

If there is a communication problem, the Linux Performance component routines automatically attempt to re-send the files. However, for the following reasons you might need to *manually* re-send the data files to OS/390:

- If Linux has been rebooted and was "down" at the time of a retry event, the event will never be rescheduled.
- If the ftp retry period has been exceeded. The ftp retry period is defined using the parameters maxRetries and retInterval in the parameter file `/etc/drl.cfg`.

All data files for those days in which the data files were not sent, remain on the /var/epdm directory. After the communication problems has been corrected, the data files for a particular day can be sent using the `drlsend` command. The `drlsend` command deletes the files after they have been successfully transferred. Issue this command with the `date` parameter, as follows:

```
drlsend  yymmdd
```

where *yymmdd* can be found in the corresponding file names, in the `/var/epdm` directory.

Collecting data under Tivoli Decision Support for z/OS

This section describes how to collect data.

General collect job

About this task

The member `DRL180.SDRLCNTL(DRLJCOLL)` contains sample JCL for a general collect.

Another way to obtain the COLLECT JCL, is to:

Procedure

1. Select 'Tivoli Decision Support for z/OS Administration' from the Tivoli Decision Support for z/OS Primary Menu.
2. Select '3. LOGS'.
3. Select 'LINUX' from the list of logs, select the 'Utilities' pull-down, and press Enter.
4. Select '1. Collect'.
5. Type your installation-specific information in this window, and press Enter.

Results

The JCL given in Figure 35 on page 108 is the general collect job.

```
//JOBCARD
//*                                                                  *
//*  Notes:                                                          *
//*    Before you submit the job:                                    *
//*    - Check the Tivoli Decision Support for z/OS and       *
//*      DB2 data set names.                                         *
//*    - Check the DB2 subsystem name (default is DSN)               *
//*      and the Tivoli Decision Support for z/OS system      *
//*      table prefix (default is DRLSYS).                           *
//*    - Insert the correct collect statement in DRLIN               *
//*      (as described above).                                       *
//*    - Specify the name of the log data set (or GDG) in            *
//*      DRLLOG. In this example, all existing generations           *
//*      are to be collected for files from Linux                    *
//*                                                                  *
//******************************************************************
//COLLECT  EXEC PGM=DRLPLC,PARM=('SYSTEM=DB2A SYSPREFIX=DRLSYS')
//STEPLIB  DD DISP=SHR,DSN=DRL180.SDRLLOAD
//         DD DISP=SHR,DSN=db2loadlibrary
//DRLIN    DD *

COLLECT LINUX;

//DRLLOG   DD DISP=SHR,DSN=USER1.COLLECT.EPDMLINUX
//DRLOUT   DD SYSOUT=*,DCB=(RECFM=F,LRECL=80)
//DRLDUMP  DD SYSOUT=*,DCB=(RECFM=F,LRECL=80)
/*
```

Figure 38. General COLLECT job

Chapter 10. Installing and configuring the Windows component

This supplements the procedure in the *Administration Guide and Reference* for installing a component, with information specific to the Windows component.

The topic describes how to plan, install, and test the Windows component.

Planning the implementation process

About this task

Before installing the Windows component, you should follow these steps to plan the implementation process:

Procedure

1. Describe user tasks. Then determine what data the Windows component must gather to help users accomplish those tasks.
2. Determine which Windows subcomponents you must install to meet the user needs.
3. Determine the administration tasks you must perform for the selected subcomponents, and make any decisions required by these tasks. These tasks help you customize Tivoli Decision Support for z/OS and the Windows component to work efficiently and effectively with your computer system.

4. Determine (for each selected subcomponent) the tasks you must perform to customize the supported products to work with Tivoli Decision Support for z/OS and with the Windows component.

Results

If this is your first exercise in implementation planning, *follow all these steps* to ensure that the Windows component's implementation is consistent. If you are reading this chaptertopic in preparation for modifying your system, you might not need to perform all of these tasks.

Use the planning process to prepare for these main customization tasks:

- Customizing Windows to generate the data required by the subcomponents you install.
- Defining environment data, which is all the information (in addition to the input data) that the Windows component needs to create reports. Environment data controls the data-collection process and provides more information in the reports.

Figure 39 illustrates the process for implementing the Windows component.

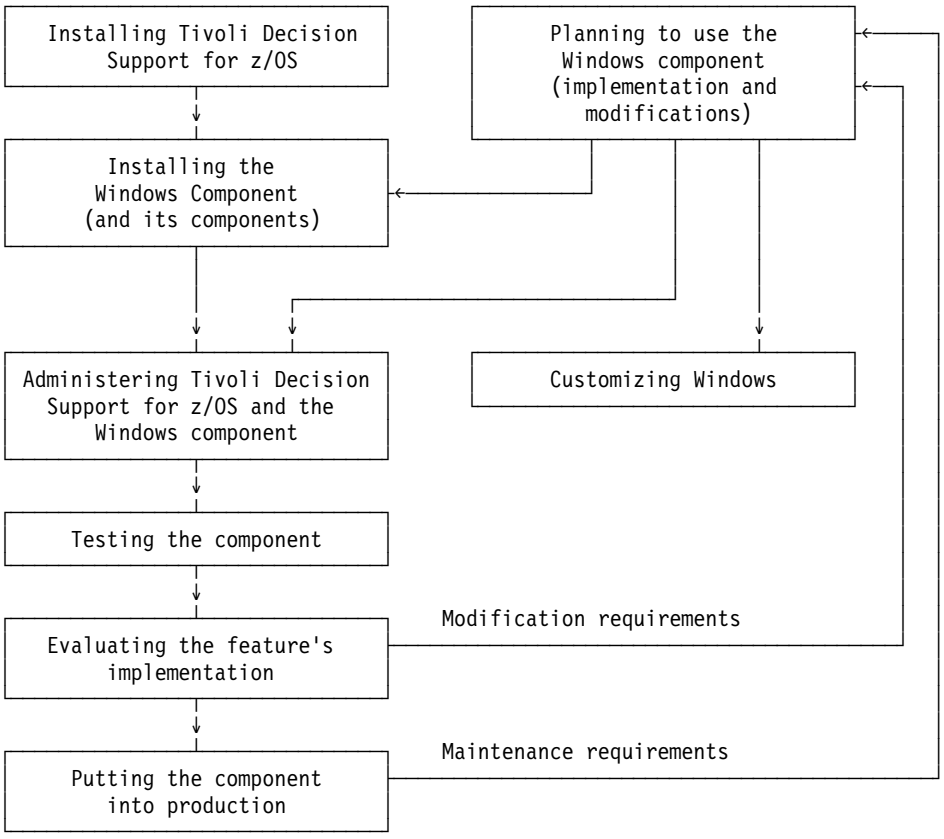


Figure 39. Implementation process for the Windows component

Considering which Windows subcomponents to install

Your most critical planning task is determining what information users need from the Windows component. For example, users may be interested only in processor information. Installing only those subcomponents needed to meet user requirements ensures that the feature benefits users while it minimizes the performance impact caused by data collection and interpretation activities.

The Windows component is divided into two subcomponents:
- Windows Performance
- Windows Device

Subcomponents are groups of Tivoli Decision Support for z/OS *objects* (for example, predefined update definitions, data tables, and reports). If you find that you need reports from a subcomponent that you have not installed, you must install that subcomponent and then wait until enough data is collected to create reports. However, if you install more subcomponents than you need, Tivoli Decision Support for z/OS collects needless data, which takes up disk space and uses processor time.

At this point, you might find it helpful to examine the predefined reports for each subcomponent, by turning to Chapter 14, "Reports," on page 151.

Installing the Windows component on your Windows nodes

This section gives you practical step-by-step information on how to install Tivoli Decision Support for z/OS on your Windows nodes.

Step 1: Check Windows requirements

About this task

Check the following Windows requirements:
- "Disk space requirements"
- "Software requirements" on page 120

Disk space requirements

You need the following approximate disk space on each Windows node:

Table 20. Disk space requirements on each Windows node

Installation task	Disk space (approximate)
Loading code in the file system, e.g., `c:\tdszwin\bin`:	1MB
Creating configuration file in, for example, `c:\tdszwin\etc`:	1MB
Creating directory for data files in, for example, `c:\tdszwin\var`:	10MB (see table note)
Note: The 10MB is based on the following assumptions: • Performance information written every 5 minutes (default) resulting in about 12 x 24 x 100 bytes (30K) per day. • Device information written every 30 minutes (default) resulting in about 2 x 24 x 100 bytes (5K) per day. • Allowance for a couple of hundred days of these files plus some overhead.	

Software requirements

The Windows component requires all of the following programs:

- Microsoft Windows 2003 Server or 2008 Server
- WMI enabled
- VBScript enabled

Step 2: Transfer Tivoli Decision Support for z/OS feature code to Windows

About this task

The Windows part of the Windows component is distributed in DRL*xxx*.SDRLWS(DRLWIN). Download this zip file using these steps:

Procedure

1. Log in to the Windows node and open a command window.
2. Enter the following commands (we will assume all the files are to be stored under *c:\tdszwin* although you can place them anywhere as long as you adjust paths in later instructions: *bin* is for program binaries, *etc* for configuration and *var* for data files):

```
c:
mkdir \tdszwin
mkdir \tdszwin\bin
mkdir \tdszwin\etc
mkdir \tdszwin\var
cd \tdszwin\bin
```

3. Start an `ftp` session to your OS/390 host, and receive the file from the OS/390 user where you installed Tivoli Decision Support for z/OS. In the following example, the highlighted entries show the commands you must enter. Note that:

   yourmvs
   is the IP address/name of the OS/390 system where Tivoli Decision Support for z/OS was installed and where the tar file is located.

   epdmwin
   is the TSO userid that has read access to the tar file.

```
c:\tdszwin\bin> ftp yourmvs
Connected to yourmvs.yourcorp.com.
220-FTPSERVE at YOURMVS.SE.IBM.COM, 11:05:24 on 03/24/00
220 Connection will close if idle for more than 5 minutes.
Name (yourmvs:root): epdmwin
331 Send password please.
Password: secret
230 EPDMWIN is logged on.
ftp> binary
200 Representation type is IMAGE.
ftp> get 'drl181.sdrlws(drlwin)' epdmwin.zip
200 Port request OK.
125 Sending data set DRL181.SDRLWS(DRLWIN) FIXrecfm 128
10240 bytes received in 0.05 seconds (204.80 Kbytes/s)
ftp> quit
221 Quit command received. Goodbye.
c:\tdszwin\bin>
```

4. On the Windows node, use Windows Explorer to unzip the contents of epdmwin.zip into your *bin* directory.

5. (Optional) After completing the previous step you can remove the zip file. (If you need the zip file in the future, it still exists on OS/390 and can be used again).
6. A sample drl.cfg parameter file is distributed as c:\tdszwin\bin\drl.cfg-sample. Copy and rename the parameter file to c:\tdszwin\etc\drl.cfg using Windows Explorer.

Capturing Windows component data and transmitting to OS/390

About this task

This section describes:

- The steps you must perform to set up the Windows component system to capture performance data, and to transmit the data to the OS/390 system:
  Step 1: Customize Windows node.
  Step 2: Set up scheduled tasks.

Step 1: Customize Windows node

About this task

Edit the parameter file `c:\tdszwin\etc\drl.cfg` to change the default values supplied by the Windows component to those for your installation.

Parameter file values to change

```
NodeName     = IBM-BOX01
XmitServer   = pthbox3.au.ibm.com
XmitUser     = drluser
XmitPassword = drlpwd
DrlPerfdXlat = 'drldata.perfd.n{n5,5}.d{f12,6}'
```

Parameter
Description

NodeName
The unique identifier for this node (up to 16 characters).

XmitServer
Host name (TCP nodename or TCP internet address) of the z/OS host to receive the Windows component data files.

XmitUser
User name to use for transmitting to host.

XmitPassword
Password to use for transmitting to host. This drl.cfg file should be protected as much as practicable by the Windows operating system. In addition, the RACF user should be set up with only enough capabilities to write to the required locations.

DrlPerfdXlat
The translation string used to form the host file name. The characters in this translation string are copied as is except when one of the following constructs is found (X and Y are considered to be numbers).
{nX,Y} will cause characters from the NodeName to be copied to the file name. The X indicates start position (1 is the first character) and Y indicates the number of characters to copy.
{fX,Y} will cause characters from the input file name to be copied to the file name. The X indicates start position (1 is the first character) and Y indicates the number of characters to copy.

So, with the node name of "IBM-PAXPRIM" and input file name of "drlperfd_20090529.txt", the translation string "'tdssrv.log.n{n5,7}.d{f12,6}'" will result in the file 'tdssrv.log.nPAXPRIM.d090529' being created on the host. Any non-valid characters in the resulting name will be replaced with "$".

Step 2: Set up scheduled tasks

About this task

The processes that will collect performance and disk space data during the day, are initiated by Windows Scheduled Tasks.

The process which transmits the information to the host is also controlled by Windows Scheduled Tasks.

You can use other scheduling solutions, such as Tivoli Workload Scheduler, but they are not covered here.

In order to set up `drlperfd`, you should (for Windows 2003 Server):

Procedure

1. From the "Start" menu, add a scheduled task under the control panel (the method to do this may vary depending on how your operating system is set up).
2. From the scheduled task wizard, click the "Next" button.
3. Click on "Browse" and select "c:\tdszwin\bin\drlperfd.vbs", then click on the "Open" button.
4. Call the task "drlperfd", choose "Daily" then click on the "Next" button.
5. Set the start time to "12:02 AM", every day, starting today then click on the "Next" button.
6. Enter your workstation password twice then click on the "Next" button.
7. Check the "Open advanced properties" then click on the "Finish" button.
8. Insert "wscript " (note space at end) before the current run text.
9. Append "c:\tdszwin\etc\drl.cfg c:\tdszwin\var" to the end of the run text.
   For example:
   ```
   wscript c:\tdszwin\bin\drlperfd.vbs c:\tdszwin\etc\drl.cfg c:\tdszwin\var
   ```
10. Click on the "OK" button and re-enter the password if necessary.
11. Exit from the scheduled task control panel application.

Results

Set up a similar task called `drlperfd_boot`, the only difference being that it should be scheduled to start when the computer starts.

Set up a task for `drlxmit.vbs` to start at 12:10AM, 12:20AM and 12:30AM.

From scheduled tasks, run the drlperfd task by right-clicking on it and selecting "Run". You should find that the control file and log file are created in c:\tdszwin\var. Once a suitable time has passed (enough to reach a 5-minute boundary like 10:00, 10:05 or 12:35), the data file will also appear.

Installing Windows component on the z/OS system

About this task

When you install Windows component subcomponents, Tivoli Decision Support for z/OS will install the required log and record definitions, record procedures, and update definitions to Tivoli Decision Support for z/OS system tables. Tivoli Decision Support for z/OS will also install the predefined tables (described in Chapter 12, "Data, lookup, and control tables," on page 133) and reports (described in Chapter 14, "Reports," on page 151). To install the Windows subcomponents, use the Administration dialog.

Perform steps 1 to 4, as follows:

Procedure

1. From the Tivoli Decision Support for z/OS Administration window (Figure 40), select 2, Components and press Enter.

```
  Other  Utilities  Help
-------------------------------------------------------------------------------
                  Tivoli Decision Support for OS/390 Administration

Select one of the following.  Then press Enter.


              2_  1.  System
                  2.  Components
                  3.  Logs
                  4.  Tables
                  5.  Reports









Command ===> _____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F9=Swap     F10=Actions  F12=Cancel
```

Figure 40. Tivoli Decision Support for z/OS Administration window

The Components window is displayed, (as shown in Figure 41 on page 124).

```
  Components  Other  Help
-------------------------------------------------------------------------------
                                 Components                ROW 1 TO  13 OF 47

Select one or more components. Then press Enter to Open component.

/   Components                                              Status     Date
_   Network NetView FTP Component                           Installed  00-03-19
_   Network NPM Internal Utilization Component              Installed  00-03-19
_   Network NV/SM Internal Utilization Component            Installed  00-03-19
_   Network Problem Component                               Installed  00-03-19
_   Network RTM Response Time Component                     Installed  00-03-19
_   Network Service Component                               Installed  00-03-19
_   DB2 Component                                           Installed  00-03-17
_   Information/Management Component                        Installed  00-03-17
/   Windows Component

Command ===> ____________________________________________________________________
 F1=Help      F2=Split     F3=Exit      F5=New       F6=Install   F7=Bkwd
 F8=Fwd       F9=Swap     F10=Actions  F12=Cancel
```

Figure 41. Components window

2. From the Components window, select the components to install (here, the Windows component) and press F6.
3. If individual subcomponents are listed, select the desired subcomponents for installation.
4. The Installation Options window is displayed, (as shown in Figure 42).

```
  Components  Other  Help
.-------------------------------------------------------------------------------.
|                                                                 ROW 1 TO 7 OF 7 |
|  .-------------------------------------------------------------.              |
| S|                 Installation Options                        | Enter.       |
|  |                                                             |              |
| /| Select one of the following.  Then press Enter.             |Date          |
| /|                                                             |              |
| _|     __  1. Online                                          |              |
| _|         2. Batch                                           |              |
| _|                                                            |              |
| _| F1=Help     F2=Split    F6=Objects  F9=Swap    F12=Cancel   |              |
| _|------------------------------------------------------------|              |
| _                                                                             |
| ***************************** BOTTOM OF DATA ******************************  |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
| Command ===> ____________________________________________________________     |
|  F1=Help      F2=Split    F7=Bkwd     F8=Fwd      F9=Swap    F12=Cancel      |
'-------------------------------------------------------------------------------'
F18=Fwd       F9=Swap     F10=Actions  F12=Cancel
```

Figure 42. Installation Options window

5. Using the component-installation procedure in the *Administration Guide and Reference* specify whether the component is to be installed online, or in batch mode.

   Batch mode installation results in less output than in online. Furthermore, for online installation your terminal will be blocked for the duration of the installation. Therefore, it is recommended that you install components in batch.

Collecting data under Tivoli Decision Support for z/OS

There are various methods you can define yourself, to set up the JCL for the COLLECT procedure. This section, however, describes a method that uses a normal data set. It explains the JCL for a general *collect* job, which uses a data set to collect data into Tivoli Decision Support for z/OS tables.

General collect job

The member DRL*xxx*.SDRLCNTL(DRLJCOLL) contains sample JCL for a general collect.

Another way to obtain the COLLECT JCL, is to:

1. Select 'Tivoli Decision Support for z/OS Administration' from the Tivoli Decision Support for z/OS Primary Menu.
2. Select '3. LOGS'.
3. Select 'Windows' from the list of logs, select the 'Utilities' pull-down, and press Enter.
4. Select '1. Collect'.
5. Type your installation-specific information in this window, and press Enter.

Figure 43 shows an example of the general collect job.

```
//JOBCARD
//*                                                              *
//*  Notes:                                                      *
//*    Before you submit the job:                                *
//*    - Check the Tivoli Decision Support for z/OS and DB2   *
//*      data set names.                                         *
//*    - Check the DB2 subsystem name (default is DSN)           *
//*      and Tivoli Decision Support for z/OS                 *
//*      system table prefix (default is DRLSYS).                *
//*    - Insert the correct collect statement in DRLIN           *
//*      (as described above).                                   *
//*    - Specify the name of the log data set (or GDG) in        *
//*      DRLLOG. In this example, all existing generations       *
//*      are to be collected for files from Windows                 *
//*                                                              *
//****************************************************************
//COLLECT  EXEC PGM=DRLPLC,PARM=('SYSTEM=DB2A SYSPREFIX=DRLSYS')
//STEPLIB  DD DISP=SHR,DSN=DRL180.SDRLLOAD
//         DD DISP=SHR,DSN=db2loadlibrary
//DRLIN    DD *

COLLECT WINDOWS;

//DRLLOG   DD DISP=SHR,DSN=USER1.COLLECT.EPDMWIN
//DRLOUT   DD SYSOUT=*,DCB=(RECFM=F,LRECL=80)
//DRLDUMP  DD SYSOUT=*,DCB=(RECFM=F,LRECL=80)
/*
```

Figure 43. General COLLECT job

Testing the installation

About this task

Before starting the daily use of the Windows component, run a few tests to check that:

Procedure

1. The installation was successful.
   - Tivoli Decision Support for z/OS is collecting the correct data
   - The data is being stored correctly
   - The correct data is being used for the creation of reports
2. The lookup tables contain appropriate values.

Results

Refer to the *Administration Guide and Reference* for detail about the steps involved in testing component installation.

Putting the feature into production

After you run the tests and verify that the installation is successful, you can put the Windows component and its subcomponents into production.

Figure 44 shows the daily steps involved in using Tivoli Decision Support for z/OS.

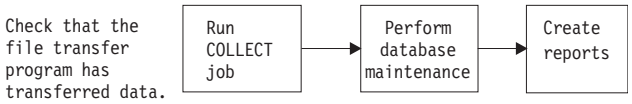


Figure 44. Daily steps involved in using Tivoli Decision Support for z/OS

You can run reports in batch, after setting batch parameters for each report using the administration dialog.

For detailed information about these steps, refer to the *Administration Guide and Reference*.

Part 4. Distributed Systems Performance Feature Reference (heritage)

Chapter 11. Data flow and Tivoli Decision Support for z/OS objects

This chaptertopic describes:

- The general data flow, starting with the gathering of data at the UNIX, Linux and Windows nodes into log files, and ending with the production of Tivoli Decision Support for z/OS reports.
- The record definitions and log types used for mapping data
- The data flow for these Distributed Systems Performance subcomponents, including the names of log files, and Tivoli Decision Support for z/OS records, tables, and reports:
  - Accounting subcomponent
  - Configuration subcomponent
  - Error subcomponent
  - Performance subcomponent

Distributed Systems Performance component general data flow

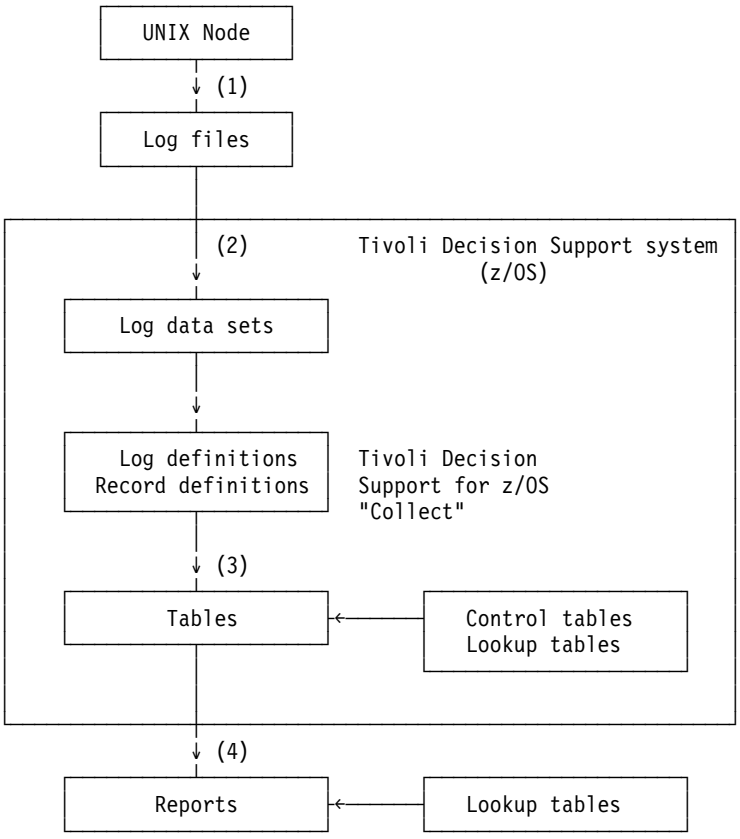


Figure 45. General Distributed Systems Performance component data flow

The processing steps shown in Figure 45 are:

1 Agents *gather* data into log files

2 Transmit log file to Tivoli Decision Support for z/OS.

3 *Collect* log data set information into Tivoli Decision Support for z/OS tables.

4 Create reports.

The following sections explain these steps in more detail.

1. **Gather data**, Step 1

   The basic data used by the Distributed Systems Performance component is *gathered* by agents into log files. Agents either issue commands or process files, to gather the data. For each distributed component, the method of gathering data into log files is:

   Component
   How data is gathered

   UNIX Accounting
   Agents process UNIX files pacct, wtmp, dtmp, qacct

   UNIX Configuration
   Agents issue AIX commands lsdev, lslpp

   UNIX Error
   Agents issue AIX command errpt

   UNIX Performance
   Agents issue AIX commands df, lsvg, lsps, and UNIX commands iostat, vmstat

   Windows Performance
   Windows Management Instrumentation (WMI)

   Windows Device
   Windows Management Instrumentation (WMI)

   Log files and their use in creating Tivoli Decision Support for z/OS records, are described in Table 21 on page 131.

2. **Transmit log file to Tivoli Decision Support for z/OS**, Step 2

   To transmit the log file to Tivoli Decision Support for z/OS, TCP/IP is used.

3. **Collect log data set information to Tivoli Decision Support for z/OS tables**, Step 3

   In processing *each record contained in the log data set*, the collect procedure:

   a. Uses a *log collector* to update the Tivoli Decision Support for z/OS table with the record. To do this, the log collector:
      - Uses the log definition and record definitions to update the Tivoli Decision Support for z/OS table with the reformatted record.
      - Uses an *update definition* to decide which reformatted record fields are to be included in which Tivoli Decision Support for z/OS table, including further summarizing into other tables (for example, updating the monthly table XPERF_DISK_M from the information used for updating the daily table XPERF_DISK_D).
      - Takes information from *control tables* (for example the UNIX Performance component may determine the period in which the measurements were made by looking up the day type information in the SPECIAL_DAY or DAY_OF_WEEK tables).

- Uses *lookup tables* (which contain user-defined information that defines an organization's operating environment) to add user-defined data to the Tivoli Decision Support for z/OS table record.

A description of the *collect* procedure is provided in the *Language Guide and Reference*, SH19-6817.

A description of the use of control tables is provided in the *Administration Guide and Reference*, SH19-6816.

4. **Create reports**, Step 4

   A description of how to create new reports is provided in the *Guide to Reporting*.

   The reports that can be created when using the UNIX Performance component are described in Chapter 14, "Reports," on page 151.

Description of log files, record and log definitions

Table 21 gives for each Distributed Systems Performance subcomponent, the node log file (and record sub-types, if any), the Tivoli Decision Support for z/OS record definition, and the Tivoli Decision Support for z/OS log definition.

Table 21. Record and log definitions used by the Distributed Systems Performance component

Performance component subcomponent	Log file (and record sub-types, when used)	Tivoli Decision Support for z/OS record definition and description (see table note 2)	Tivoli Decision Support for z/OS log definition
UNIX Accounting	PACCT (command accounting)	XACCT_COMMAND (command accounting data)	UNIX
UNIX Accounting	WTMP (connect time accounting)	XACCT_CONNECT (user accounting data)	UNIX
UNIX Accounting	DTMP (disk accounting)	XACCT_DISK (disk usage data)	UNIX
UNIX Accounting	QACCT (print accounting)	XACCT_PRINT (print accounting data)	UNIX
UNIX Configuration	CONF (A,D,S) (configuration hardware)	XCONFIG_HW (configuration of hardware data)	UNIX
UNIX Configuration	CONF (H,I,L) (configuration software)	XCONFIG_SW (configuration of software data)	UNIX
UNIX Error	ERRPT (error)	XERROR (error log data)	UNIX
UNIX Performance	PERF (VM) (processor performance)	XPERF_CPU (processor usage data)	UNIX
UNIX Performance	DISK (disk performance)	XPERF_DISK (disk space usage data)	UNIX
UNIX Performance	PERF (IO) (disk I/O performance)	XPERF_DISKIO (disk I/O data)	UNIX
UNIX Performance	PERF (PS) (paging performance)	XPERF_PAGING (paging space usage data)	UNIX
Windows Performance	NA	WINDOWS_PERF (Windows Performance)	WINDOWS_LOG

Table 21. Record and log definitions used by the Distributed Systems Performance component (continued)

Performance component subcomponent	Log file (and record sub-types, when used)	Tivoli Decision Support for z/OS record definition and description (see table note 2)	Tivoli Decision Support for z/OS log definition
Windows Device	NA	WINDOWS_DISK (Windows Disk)	WINDOWS_LOG

Note:

- Log file names also contain the name of the UNIX node, and the date on which the log file was created.
- The fields contained in each Tivoli Decision Support for z/OS record definition are described in Chapter 15, "Log record definitions," on page 201. You can also display these fields, when you do the following:
  1. Select 2 (Tivoli Decision Support for z/OS Administration), from the Tivoli Decision Support for z/OS Primary Menu.
  2. Select 3 (Logs).
  3. Select the log definition UNIX.
  4. Select the record definition you require, and the fields will be displayed.

The Distributed Systems Performance component does not use Tivoli Decision Support for z/OS record procedures.

Chapter 12. Data, lookup, and control tables

The Tivoli Decision Support for z/OS database is a collection of DB2 tables, where each table contains a fixed number of columns. The number of rows in each table varies with time, because of rows added by the collect function and because of database maintenance.

This topic describes:

- The format that is used for defining Distributed Systems Performance component table names and views
- The layout of the tables
- The tables used by the UNIX accounting subcomponent
- The table used by the UNIX configuration subcomponent
- The tables used by the UNIX error subcomponent
- The tables used by the UNIX performance subcomponent
- The tables used by the Windows component
- The lookup tables used by some of the subcomponents

Note: For descriptions of common data tables used by the Distributed Systems Performance component and other Tivoli Decision Support for z/OS features, refer to the *Administration Guide and Reference*, SH19-6816.

Naming standard for tables

The names of the Distributed Systems Performance component tables use this format:

X*subcomponent_content_suffix* or ***WIN_PERF_subcomponent_suffix***

where:

- ***subcomponent*** identifies the subcomponent to which the table belongs (for example, ***ACCT*** for the accounting subcomponent).
- ***content*** is a description (for example, XACCT_***COMMAND*** for the accounting subcomponent command statistics).
- ***suffix*** indicates the summarization level of the data in the table (for example, XACCT_COMMAND_***D*** for command statistics summarized by day). Table names for the configuration subcomponent do not contain suffixes.

  A table name can have these summarization-level suffixes:

  _H The table holds data summarized by **hour** (hourly data).
  _D The table holds data summarized by **day** (daily data).
  _M The table holds data summarized by **month** (monthly data).

  Lookup tables and control tables do not have a suffix; control tables also do not have a prefix.

The names of Distributed Systems Performance component *views* use the format consisting of the table name from which the view is taken, followed by a *V*. For example, the view XACCT_DISK_MV is taken from the table XACCT_DISK_M.

Table descriptions

Each table description includes information about the table, a description of each of the *key columns*, and a description of each of the *data columns*.

- Key columns are marked with a "K". They are sorted in the sequence they appear in the table.
- Data columns follow the last key column and are sorted in alphabetical order with the underscore ignored.

The descriptions of most key columns and data columns contain references to the fields from which they are derived in the record (for example, "From AC_UID"). For an explanation of such fields, refer to the applicable product documentation.

For each subcomponent, the tables appear in alphabetical order, with underscores and suffixes ignored.

Tables with similar contents (that is, tables with the same name but with different suffixes) are described under one heading. For example, the heading "XACCT_COMMAND_D, _M" covers two similar tables: XACCT_COMMAND_D and XACCT_COMMAND_M. Except for the DATE column, the contents of these tables are identical. Differences that exist in the contents of similar tables are explained in the column descriptions.

Tables in the UNIX accounting subcomponent

This section describes the accounting subcomponent tables:

XACCT_COMMAND_D, _M

These tables provide daily and monthly statistics on UNIX commands, and resource consumption. They contain command accounting data from UNIX records with record type 'PACCT'.

The default retention periods are:
30 days for XACCT_COMMAND_D
765 days for XACCT_COMMAND_M

Column name		Data type	Description
DATE	K	DATE	Date when the records were written. For XACCT_COMMAND_M, this is the first day of the month. From DTE.
NODE_NAME	K	CHAR(16)	Node name. From NODE_NAME.
PERIOD_NAME	K	CHAR(8)	Name of the period. This is derived using fields NODE_NAME, DTE and TME from the record as parameters in the PERIOD function.
USER_GROUP	K	CHAR(8)	Group name. From AC_GID.

Column name		Data type	Description
NODE_GROUP	K	CHAR(16)	Node group. From ACCOUNTING_GROUP in the X_NODE_NAME lookup table. This is derived using field NODE_NAME from the record as key. If no match is found, this column is set to '?'.
USER_NAME	K	CHAR(8)	User name. From AC_UID.
TTY	K	CHAR(8)	Control typewriter. From AC_TTY.
COMMAND	K	CHAR(12)	Command name. From AC_COMM.
COMMANDS		INTEGER	Total number of commands. This is the count of records.
ELAPSED_SEC		FLOAT	Total elapsed time, in seconds. Calculated as the sum of AC_ETIME/64.
IO_CHARS		INTEGER	Total characters transferred. This is the sum of AC_IO.
MEMORY_BYTES		FLOAT	Total memory usage, in bytes. This value should be divided by COMMANDS to get the average memory usage. This is the sum of AC_MEM.
MEMORY_KB_SEC		FLOAT	Total memory usage, in kilobyte-seconds. Calculated as the sum of (AC_MEM/1024)*(AC_ETIME/64).
RW_BLOCKS		INTEGER	Total blocks read/written. This is the sum of AC_RW.
SYSTEM_SEC		FLOAT	Total system time, in seconds. Calculated as the sum of AC_STIME/64.
USER_SEC		FLOAT	Total user time, in seconds. Calculated as the sum of AC_UTIME/64.

XACCT_DISK_D, _M

These tables provide daily and monthly statistics on UNIX disk usage. They contain disk accounting data from UNIX records with record type 'DTMP'.

The default retention periods are:
30 days for XACCT_DISK_D
765 days for XACCT_DISK_M

Column name		Data type	Description
DATE	K	DATE	Date when the records were written. For XACCT_DISK_M, this is the first day of the month. From DTE.
NODE_NAME	K	CHAR(16)	Node name. From NODE_NAME.
USER_NAME	K	CHAR(8)	User name. From USER_NAME.
NODE_GROUP	K	CHAR(16)	Node group. From ACCOUNTING_GROUP in the X_NODE_NAME lookup table. This is derived using field NODE_NAME from the record as key. If no match is found, this column is set to '?'.
DISK_BLOCKS_AVG		FLOAT	Average number of disk blocks. This is the average of DISK_BLOCKS over RECORDS_COLLECTED.
DISK_BLOCKS_SUM		FLOAT	Total number of disk blocks. This is the sum of DISK_BLOCKS. Before using this column, divide it by the value in the RECORDS_COLLECTED column.
RECORDS_COLLECTED		INTEGER	Number of records. This is the count of records.

XACCT_PRINT_D, _M

These tables provide daily and monthly statistics on UNIX printing. They contain print accounting data from UNIX records with record type 'QACCT'.

The default retention periods are:
30 days for XACCT_PRINT_D
765 days for XACCT_PRINT_M

Column name		Data type	Description
DATE	K	DATE	Date when the records were written. For XACCT_PRINT_M, this is the first day of the month. From DTE.
NODE_NAME	K	CHAR(16)	Node name. From NODE_NAME.
PERIOD_NAME	K	CHAR(8)	Name of the period. This is derived using fields NODE_NAME, DTE and TME from the record as parameters in the PERIOD function.
USER_NAME	K	CHAR(8)	User name. From the first word of PR_USER.
PRINT_QUEUE	K	CHAR(8)	Print queue name. From PRINT_QUEUE.
PRINT_TYPE	K	CHAR(8)	Print queue type. From PRINT_TYPE in the XACCT_PRINT_TYPE lookup table. This is derived using field PRINT_QUEUE from the record as key. If no match is found, this column is set to the value in field PRINT_QUEUE.
NODE_GROUP	K	CHAR(16)	Node group. From ACCOUNTING_GROUP in the X_NODE_NAME lookup table. This is derived using field NODE_NAME from the record as key. If no match is found, this column is set to '?'.
REQUEST_NODE	K	CHAR(32)	Node name of requesting node. From the second word of PR_USER. If no value is found the printout is local and this column is set to NODE_NAME.
PAGES		INTEGER	Number of pages. This is the sum of PAGES.
PRINTOUTS		INTEGER	Number of printouts. This is the count of records.

XACCT_CONNECT_D, _M

These tables provide daily and monthly statistics on UNIX users and their connections. They contain connect time accounting data from UNIX records with record type 'WTMP'.

The default retention periods are:
30 days for XACCT_CONNECT_D
765 days for XACCT_CONNECT_M

Column name		Data type	Description
DATE	K	DATE	Date when the records were written. For XACCT_CONNECT_M, this is the first day of the month.
NODE_NAME	K	CHAR(16)	Node name. From NODE_NAME.
PERIOD_NAME	K	CHAR(8)	Name of the period. This is derived using fields NODE_NAME, DTE and TME from the record as parameters in the PERIOD function.
USER_NAME	K	CHAR(8)	User login name. From UT_USER.
CONNECT_TYPE	K	CHAR(2)	Connect type code. From UT_TYPE.
CONNECTS		INTEGER	Total number of connections. This is the count of records.

XACCT_DISK_MV

This view provides monthly UNIX disk accounting statistics for the 'root' user. It is based on the XACCT_DISK_M table.

Column name		Data type	Description
DATE		DATE	Date when the record was written. From DTE. This is the date of the first day of the month.
NODE_NAME		CHAR(16)	Node name. From NODE_NAME.
RECORDS_COLLECTED		INTEGER	Number of records. This is the count of records.

Tables in the UNIX configuration subcomponent

This section describes the configuration subcomponent tables:

- XCONFIG_HW
- XCONFIG_SW

XCONFIG_HW

This table provides AIX hardware configuration data. It contains hardware configuration data from AIX records with record type 'CONF' and subtype 'A', 'D' or 'S'.

The default retention period for this table is 765 days.

Column name		Data type	Description
DATE	K	DATE	Date when the record was written. From DTE.
TIME	K	TIME	Time when the record was written. From TME.
NODE_NAME	K	CHAR(16)	Node name. From NODE_NAME.
DEVICE_CLASS	K	CHAR(16)	Device class. From DEVICE_CLASS.
DEVICE_SUBCLASS	K	CHAR(16)	Device subclass. From DEVICE_SUBC.
DEVICE_NAME	K	CHAR(16)	Device name. From DEVICE_NAME.
DEVICE_TYPE	K	CHAR(16)	Device type. From DEVICE_TYPE.
STATUS	K	CHAR(2)	Device status. From RECORD_SUBTYPE.

XCONFIG_SW

This table provides AIX software configuration data. It contains software configuration data from AIX records with record type 'CONF' and subtype 'H', 'I' or 'L'.

The default retention period for this table is 765 days.

Column name		Data type	Description
DATE	K	DATE	Date when the record was written. From DTE.
TIME	K	TIME	Time when the record was written. From TME.
NODE_NAME	K	CHAR(16)	Node name. From NODE_NAME.
SOFTWARE_OBJECT	K	CHAR(32)	Software object name. From FILE_NAME.
PATH	K	CHAR(32)	Path name for object. From FILE_NAME.
DESCRIPTION		VARCHAR(80)	Product description. From FILE_NAME.

Tables in the configuration subcomponent

Column name		Data type	Description
FEATURE		CHAR(4)	Feature ID. From FEATURE_ID.
INSTALL_DATE		CHAR(8)	Product installation date. From INST_DATE.
INSTALL_STATUS		CHAR(10)	Product installation status. From STATUS.
PRODUCT_ID		CHAR(10)	Product ID. From PRODUCT_ID.
PRODUCT_NAME		CHAR(20)	Product name. From PRODUCT_NAME.
PRODUCT_STATE		CHAR(12)	Product state. From STATE.
RELEASE		CHAR(15)	Product release number. From RELEASE.

Tables in the UNIX error subcomponent

This section describes the error subcomponent table.

XERROR_D, _M

These tables provide daily and monthly statistics on AIX error log. They contain error data from AIX records with record type 'ERRPT'.

The default retention periods are:
30 days for XERROR_D
765 days for XERROR_M

Column name		Data type	Description
DATE	K	DATE	Date when the error occurred. From DTE. For XERROR_M, this is the first day of the month.
NODE_NAME	K	CHAR(16)	Node name. From NODE_NAME.
PERIOD_NAME	K	CHAR(8)	Name of the period. This is derived using fields NODE_NAME, DTE and TME from the record as parameters in the PERIOD function.
ERROR_ID	K	CHAR(8)	Error identification. From ERROR_ID.
ERROR_TYPE	K	CHAR(1)	Error type. From ERROR_TYPE.
ERROR_CLASS	K	CHAR(1)	Error class. From ERROR_CLASS.
RESOURCE_NAME	K	CHAR(14)	Resource name. From RESOURCE_NAME.
DESCRIPTION		VARCHAR(40)	Error description. From DESCRIPTION.
ERRORS		INTEGER	Number of errors. This is the count of records.

Tables in the UNIX performance subcomponent

This section describes the performance subcomponent tables:

XPERF_CPU_H, _D, _M

These tables provide hourly, daily, and monthly statistics on UNIX utilization. They contain CPU and memory usage, paging rate and number of users and processes from UNIX records with record type 'PERF' and subtype 'VM'.

The default retention periods are:
7 days for XPERF_CPU_H
30 days for XPERF_CPU_D
765 days for XPERF_CPU_M

Column name		Data type	Description
DATE	K	DATE	Date when the records were written. For XPERF_CPU_M, this is the first day of the month. From DTE.
TIME	K	TIME	Time (rounded down to the nearest hour) when the record was written. From TME. This field is present only in XPERF_CPU_H.
NODE_NAME	K	CHAR(16)	Node name. From NODE_NAME.
PERIOD_NAME	K	CHAR(8)	Name of the period. This is derived using fields NODE_NAME, DTE and TME from the record as parameters in the PERIOD function.
CPU_USAGE_PCT_AVG		FLOAT	Average CPU usage, in percent. This is the average of CPU_AVG.
CPU_USAGE_PCT_MAX		INTEGER	Maximum CPU usage, in percent. This is the maximum of CPU_MAX.
CPU_USAGE_PCT_MIN		INTEGER	Minimum CPU usage, in percent. This is the minimum of CPU_MIN.
MEASURED_SEC		INTEGER	Total measured time, in seconds. This is the sum of INTERVAL*60.
MEM_FREE_PAGES_AVG		FLOAT	Average number of free pages. This is the average of MEMORY_AVG.
MEM_FREE_PAGES_MAX		INTEGER	Maximum number of free pages. This is the maximum of MEMORY_MAX.
MEM_FREE_PAGES_MIN		INTEGER	Minimum number of free pages. This is the minimum of MEMORY_MIN.
PAGING_RATE_AVG		FLOAT	Average number of pages paged IN/OUT per second. This is the average of PAGING_AVG.
PAGING_RATE_MAX		INTEGER	Maximum number of pages paged IN/OUT per second. This is the maximum of PAGING_MAX.
PAGING_RATE_MIN		INTEGER	Minimum number of pages paged IN/OUT per second. This is the minimum of PAGING_MIN.
PROCESSES_AVG		FLOAT	Average number of processes. This is the average of PROCESSES.
PROCESSES_MAX		INTEGER	Maximum number of processes. This is the maximum of PROCESSES.
PROCESSES_MIN		INTEGER	Minimum number of processes. This is the minimum of PROCESSES.
RECORDS_COLLECTED		INTEGER	Total number of records. This is the count of records.
USERS_AVG		FLOAT	Average number of users. This is the average of USERS.
USERS_MAX		INTEGER	Maximum number of users. This is the maximum of USERS.
USERS_MIN		INTEGER	Minimum number of users. This is the minimum of USERS.

XPERF_DISK_D, _M

These tables provide daily and monthly statistics on AIX disk usage. They contain disk performance data from AIX records with record type 'DISK' and subtype 'FS' or 'VG'.

The default retention periods are:
30 days for XPERF_DISK_D
765 days for XPERF_DISK_M

Column name		Data type	Description
DATE	K	DATE	Date when the records were written. For XPERF_DISK_M, this is the first day of the month. From DTE.
NODE_NAME	K	CHAR(16)	Node name. From NODE_NAME.
PERIOD_NAME	K	CHAR(8)	Name of the period. This is derived using fields NODE_NAME, DTE and TME from the record as parameters in the PERIOD function.
SPACE_TYPE	K	CHAR(2)	Space type, FS or VG. From RECORD_SUBTYPE.
VOLUME	K	CHAR(16)	Volume. From FILE_SYSTEM for space type FS and from VOLUME_GROUP for space type VG.
FILE_SYSTEM	K	CHAR(16)	File system. From FILE_SYSTEM for space type FS and set to blanks for space type VG.
RECORDS_COLLECTED		INTEGER	Number of samples. This is the count of records.
SPACE_FREE_MB		FLOAT	Free space, in MB, accumulated for all samples. This is the sum of FREE_SPACE_MB for space type VG and calculated as the sum of FREE_SPACE_KB/1024.0 for space type FS. This value should be divided by RECORDS_COLLECTED to get the average free space.
SPACE_SIZE_MB		FLOAT	Total size of space, in MB, accumulated for all samples. This is the sum of TOTAL_SPACE_MB for space type VG and calculated as the sum of TOTAL_SPACE_KB/1024.0 for space type FS. Divide this value by RECORDS_COLLECTED to get the average value.
SPACE_USED_PCT		FLOAT	Used space, in percent, accumulated for all samples. Calculated as the sum of 100*USED_SPACE_MB/TOTAL_SPACE_MB space type VG and as the sum of USED_SPACE_PCT for space type FS. Divide this value by RECORDS_COLLECTED to get the average value.

XPERF_DISKIO_H, _D, _M

These tables provide hourly, daily, and monthly statistics on UNIX disk I/O. They contain disk I/O performance data from UNIX records with record type 'PERF' and subtype 'IO'.

The default retention periods are:
7 days for XPERF_DISKIO_H
30 days for XPERF_DISKIO_D
765 days for XPERF_DISKIO_M

Column name		Data type	Description
DATE	K	DATE	Date when the records were written. For XPERF_DISKIO_M, this is the first day of the month. From DTE.
TIME	K	TIME	Time (rounded down to the nearest hour) when the record was written. From TME. This field is present only for XPERF_DISKIO_H.
NODE_NAME	K	CHAR(16)	Node name. From NODE_NAME.
PERIOD_NAME	K	CHAR(8)	Name of the period. This is derived using fields NODE_NAME, DTE and TME from the record as parameters in the PERIOD function.
DISK	K	CHAR(10)	Physical disk name. From DISK.
DISK_BUSY_PCT_AVG		FLOAT	Average disk busy, in percent. This is the average of BUSY.
DISK_BUSY_PCT_MAX		INTEGER	Maximum disk busy, in percent. This is the maximum of BUSY.
MEASURED_SEC		INTEGER	Total measurement time, in seconds. Calculated as the sum of INTERVAL*60.
READ_KB_TOTAL		INTEGER	Amount of data read, in kilobytes. This is the sum of TOTREAD.
READ_MAXRATE		FLOAT	Maximum amount of data read, in kilobytes per second. Calculated as the maximum of TOTREAD/(INTERVAL*60.0).
RECORDS_COLLECTED		INTEGER	Number of records. This is the count of records.
WRITE_KB_TOTAL		INTEGER	Amount of data written, in kilobytes. This is the sum of TOTWRITE.
WRITE_MAXRATE		FLOAT	Maximum amount of data written, in kilobytes per second. Calculated as the maximum of TOTWRITE/(INTERVAL*60.0).
RW_KB_TOTAL		FLOAT	Amount of data written, in kilobytes. Calculated as the sum of TOTRW.
RW_MAXRATE		INTEGER	Maximum amount of data written, in kilobytes per second. Calculated as the sum of TOTRW/(INTERVAL*60.0).

XPERF_PAGING_H, _D, _M

These tables provide hourly, daily, and monthly statistics on UNIX page space. They contain page space performance data from UNIX records with record type 'PERF' and subtype 'PS'.

The default retention periods are:
7 days for XPERF_PAGING_H
30 days for XPERF_PAGING_D
765 days for XPERF_PAGING_M

Column name		Data type	Description
DATE	K	DATE	Date when the records were written. For XPERF_PAGING_M, this is the first day of the month. From DTE.
TIME	K	TIME	Time (rounded down to the nearest hour) when the record was written. From TME. This field is present only for XPERF_PAGING_H.
NODE_NAME	K	CHAR(16)	Node name. From NODE_NAME

Column name		Data type	Description
PERIOD_NAME	K	CHAR(8)	Name of the period. This is derived using fields NODE_NAME, DTE and TME from the record as parameters in the PERIOD function.
DISK	K	CHAR(10)	Physical disk name. From DISK.
PAGE_SPACE	K	CHAR(10)	Page space name. From PAGE_SPACE.
PS_SIZE_AVG_KB		FLOAT	Average page space size, in KB. This is the average of PSSIZE.
PS_USED_AVG_KB		FLOAT	Average page space used, in KB. This is the average of PSUSE.
RECORDS_COLLECTED		INTEGER	Number of records. This is the count of records.

XPERF_PS_INFO

This table provides physical size data for the volumes used. It maps the the XPERF_PS record for LINUX systems and the XPERF_PAGING record for UNIX systems.

Column name		Data type	Description
DATE		DATE	The date when the record was written. From DTE.
NODE_NAME		CHAR(16)	The node identification.
PERIOD_NAME		CHAR(8)	The name of the period.
PHY_VOL		CHAR(20)	The name of the physical volume.
VOL_SIZE		INTEGER	The size of the volume.
BLOCKS		INTEGER	The number of blocks that are used in the volume.
SYSTEM		CHAR(5)	UNIX or LINUX system.

XPERF_VM_INFO

This table provides performance data for user memory, swap memory, and CPU activity. It maps the XPERF_VM record for LINUX systems and the XPERF_CPU record for UNIX systems.

Column name		Data type	Description
DATE		DATE	The date when the record was written. From DTE.
NODE_NAME		CHAR(16)	The node identification.
PERIOD_NAME		CHAR(8)	The name of the period.
USER		INTEGER	The number of users.
PROC		INTEGER	The number of processes.
DUR_INT		INTEGER	The duration of the interval.
MEM_MIN_U		INTEGER	The minimum memory user page.
AVG_MEM_U		FLOAT	The average memory user page.
MEM_MAX_U		INTEGER	The maximum memory user page.
MEM_MIN_S		INTEGER	The minimum memory page swap.
AVG_MEM_S		FLOAT	The average memory page swap.
MEM_MAX_S		INTEGER	The maximum memory page swap.
CPU_MIN		INTEGER	The minimum CPU value.
AVG_CPU		FLOAT	The average CPU value.

Column name		Data type	Description
CPU_MAX		INTEGER	The maximum CPU value.
SYSTEM		CHAR(5)	UNIX or LINUX system.

Tables in the Windows component

WIN_PERF_DD_H, _D, _M

These tables holds hourly, daily and monthly data for the disk device records.

The default retention periods are:
10 days for WIN_PERF_DD_H
30 days for WIN_PERF_DD_D
765 days for WIN_PERF_DD_M

Column	Key	Datatype	Description
DATE	K	DATE	The date of the record. For the _M table, dates are stored as the first day of the month.
TIME	K	TIME	The time of the record. Set to the start of the hour for table _H. This column does not exist in the _D or _M tables.
NODE_NAME	K	CHAR(16)	The node name of the machine.
DEVICE_NAME	K	CHAR(8)	The device name for the device.
SAMPLES		INTEGER	Count of the records consolidated into a table row. Used for working out averages.
TOTAL_DEVICE_SIZE		FLOAT	Device size at the end of the period.
FREE_SPACE_AVG		FLOAT	Averages the free space on the device, based on SAMPLES.
FREE_SPACE_MIN		FLOAT	The minimum free space available.
FREE_SPACE_MAX		FLOAT	The maximum free space available.
PERCENT_FREE_AVG		FLOAT	Averages the free space percentage on the device, based on SAMPLES.
PERCENT_FREE_MIN		FLOAT	The minimum free space percentage
PERCENT_FREE_MAX		FLOAT	The maximum free space percentage.

WIN_PERF_PU_H, _D, _M

This table holds hourly, daily and monthly data for the performance usage records.

The default retention periods are:
7 days for WIN_PERF_PU_H
35 days for WIN_PERF_PU_D
366 days for WIN_PERF_PU_M

Column	Key	Datatype	Description
DATE	K	DATE	The date of the record. For the _M table, dates are forced back to the first day of the month.
TIME	K	TIME	The time of the record. Set to the start of the hour for table _H. This column does not exist in the _D or _M tables.

Tables in the Windows component

Column	Key	Datatype	Description
NODE_NAME	K	CHAR(16)	The node name of the machine.
SAMPLES		INTEGER	Count of the records consolidated into a table row. Used for working out actuals.
CPU_COUNT		SMALLINT	Number of CPUs and/or cores.
CPU_USAGE_AVG		FLOAT	Averages the average CPU usage. This figure is averaged based on SAMPLES.
CPU_USAGE_MIN		FLOAT	The minimum CPU usage.
CPU_USAGE_MAX		FLOAT	The maximum CPU usage.
MEMORY_SIZE		FLOAT	The total memory size at the end of the period.
MEMORY_FREE_AVG		FLOAT	Averages the average free memory. This figure is averaged based on SAMPLES.
MEMORY_FREE_MIN		FLOAT	The minimum free memory.
MEMORY_FREE_MAX		FLOAT	The maximum free memory.

Lookup tables

This section describes the lookup tables specific to the UNIX Performance component:

- XACCT_CONNECT_TYPE
- XNODE_NAME
- XACCT_PRINT_TYPE

For descriptions of common lookup tables used by the UNIX Performance component and other Tivoli Decision Support for z/OS features, refer to the *Administration Guide and Reference*, SH19-6816.

XACCT_CONNECT_TYPE

This lookup table contains descriptions of connect types.

Column name		Data type	Description
CONNECT_TYPE	K	CHAR(2)	Connect type code.
CONNECT_DESC		CHAR(16)	Connect type description.

Example of table contents

```
PROCESS  PROCESS
 CODE     TYPE
-------  ----------------
00       EMPTY
01       RUN_LVL
02       BOOT_TIME
03       OLD_TIME
04       NEW_TIME
05       INIT_PROCESS
06       LOGIN_PROCESS
07       USER_PROCESS
08       DEAD_PROCESS
09       ACCOUNTING
```

X_NODE_NAME

This lookup table contains node names and their associated accounting groups, period plans and TCP/IP address.

Column name		Data type	Description
NODE_NAME	K	CHAR(16)	Node name. This is an alias name for the node. This may contain global search characters.
ACCOUNTING_GROUP		CHAR(16)	Accounting group.
INTERNET_ADDRESS		CHAR(15)	Internet address. This may contain global search characters.
PERIOD_PLAN		CHAR(8)	Period plan. This value can be used as a parameter when using the PERIOD function looking up PERIOD_NAME.

Example of table contents

```
NODE             ACCOUNTING        PERIOD    INTERNET
NAME               GROUP            PLAN     ADDRESS
---------------  ----------------  --------  ---------------
node1            GROUP1            BASE      9.52.50.7
node2            GROUP2            PLAN1     9.52.50.8
node3            GROUP2            PLAN2     9.52.50.9
node5            GROUP3            PLAN3     9.52.50.11
%                DEFAULT           DEFAULT   9.99.99.99
```

XACCT_PRINT_TYPE

This lookup table converts print-queues to a print-type. It updates the XACCT_PRINT_D table.

Column name		Data type	Description
NODE_NAME	K	CHAR(16)	Node name. This may contain global search characters.
PRINT_QUEUE	K	CHAR(8)	Print queue name. This may contain global search characters.
PRINT_TYPE		CHAR(8)	Print queue type.

Example of table contents

```
NODE             PRINT     PRINT
NAME             QUEUE     TYPE
---------------  --------  --------
node1            prt01     dep01
node2            prt01     dep01
node3            prt05     dep02
%                prt99     dep99
```

Lookup tables

Chapter 13. Linux data tables

This chapter describes the data tables used by the UNIX Performance component.

XCONF_HARDWARE

This table provides hardware configuration data for the devices used. It maps the DRLTXCNF record.

Column name		Data type	Description
DATE		DATE	The date when the record was written. From DTE.
NODE_NAME		CHAR(16)	The node identification.
PERIOD_NAME		CHAR(8)	The name of the period.
DEVICE_CLS		CHAR(10)	The device class of the hardware element (Video, HD, CD).
CPU		CHAR(10)	The CPU identification.
ADD_ID		CHAR(10)	The address ID of the UNIX device.
DEVICE_ASG		CHAR(7)	The type of device used in the UNIX system.
BUS		CHAR(7)	The type of bus used in the UNIX system.

XCONF_SOFTWARE

This table provides software configuration data for the packages used. It maps the DRLTXCNF record.

Column name		Data type	Description
DATE		DATE	The date when the record was written. From DTE.
NODE_NAME		CHAR(16)	The node identification.
PERIOD_NAME		CHAR(8)	The name of the period.
PACK_NAME		CHAR(50)	The name of the package used on the system.
SOFT_VER		CHAR(12)	The software version of the package installed on the system.
INST_TS		CHAR(12)	The timestamp of when the software was installed.

XDISK_INFOR

This table provides space allocation data for the disks on the file system. It maps the DRLTXDSK record.

Column name		Data type	Description
DATE		DATE	The date when the record was written. From DTE.
NODE_NAME		CHAR(16)	The node identification.
PERIOD_NAME		CHAR(8)	The name of the period.
FILE_SYS		CHAR(9)	The name of the file system.
TOT_SIZE		INTEGER	The maximum size of the disk.
FREE		INTEGER	The free space on the disk.

XDTMP_INFOR

This table provides process information related to users. It maps the DRLTXDTM record.

Column name		Data type	Description
DATE		DATE	The date when the record was written. From DTE.
NODE_NAME		CHAR(16)	The node identification.
PERIOD_NAME		CHAR(8)	The name of the period.
US_ID		CHAR(9)	The user ID.
US_NAME		CHAR(20)	The user name.
USED_BLOCK		CHAR(10)	The number of disk blocks that are used.

XPERF_PS_INFO

This table provides physical size data for the volumes used. It maps the XPERF_PS record for Linux systems and the XPERF_PAGING record for UNIX systems.

Column name		Data type	Description
DATE		DATE	The date when the record was written. From DTE.
NODE_NAME		CHAR(16)	The node identification.
PERIOD_NAME		CHAR(8)	The name of the period.
PHY_VOL		CHAR(20)	The name of the physical volume.
VOL_SIZE		INTEGER	The size of the volume.
BLOCKS		INTEGER	The number of blocks that are used in the volume.
SYSTEM		CHAR(5)	UNIX or Linux system.

XPERF_VM_INFO

This table provides performance data for user memory, swap memory, and CPU activity. It maps the XPERF_VM record for Linux systems and the XPERF_CPU record for UNIX systems.

Column name		Data type	Description
DATE		DATE	The date when the record was written. From DTE.
NODE_NAME		CHAR(16)	The node identification.
PERIOD_NAME		CHAR(8)	The name of the period.
USER		INTEGER	The number of users.
PROC		INTEGER	The number of processes.
DUR_INT		INTEGER	The duration of the interval.
MEM_MIN_U		INTEGER	The minimum memory user page.
AVG_MEM_U		FLOAT	The average memory user page.
MEM_MAX_U		INTEGER	The maximum memory user page.
MEM_MIN_S		INTEGER	The minimum memory page swap.
AVG_MEM_S		FLOAT	The average memory page swap.
MEM_MAX_S		INTEGER	The maximum memory page swap.

Column name		Data type	Description
CPU_MIN		INTEGER	The minimum CPU value.
AVG_CPU		FLOAT	The average CPU value.
CPU_MAX		INTEGER	The maximum CPU value.
SYSTEM		CHAR(5)	UNIX or Linux system.

XWTMP_INFOR

This table provides process information related to users. It maps the DRLTXDTM record.

Column name		Data type	Description
DATE		DATE	The date when the record was written. From DTE.
NODE_NAME		CHAR(16)	The node identification.
PERIOD_NAME		CHAR(8)	The name of the period.
USER_LOG		CHAR(8)	The user login name.
ENTRY		CHAR(2)	The entry type.
PROC_ID		CHAR(6)	The process ID.
NODE		CHAR(16)	The node name.

Chapter 14. Reports

The reporting function produces reports based on the data in the Tivoli Decision Support for z/OS database. Reports can show data from tables or from views. You can request reports online or by submitting batch jobs. Typically, you use online reporting for reports that you use once, and batch reporting for regularly required reports.

This topic describes:

- The format of the names used to define each report, and how source tables, attributes and variables are used.
- The reports in the accounting subcomponent
- The reports in the configuration subcomponent
- The reports in the error subcomponent
- The reports in the performance subcomponent

Report format and general description

Tivoli Decision Support for z/OS presents reports in tables and graphs. All reports have the same basic report layout. This section describes the elements that are common among Tivoli Decision Support for z/OS feature reports:

- Report ID
- Report group
- Source
- Attributes
- Variables

Report ID

Tivoli Decision Support for z/OS assigns each report a unique identifier. The UNIX Performance component uses this format for report IDs:

Xsubcomponentxx

Where:

subcomponent
: Identifies the subcomponent to which the report belongs. It can be one of the following:
  ACCT The accounting subcomponent.
  CONFIG
  The configuration subcomponent.
  ERROR
  The error subcomponent.
  PERF The performance subcomponent.

xx
: A sequential number identifying the report.

Examples:

XACCT01
XPERF02

Report group

Tivoli Decision Support for z/OS uses several predefined report groups. For the UNIX Performance component, each subcomponent has one group. The four UNIX Performance component report groups are given in "Report groups" on page 82.

Source

Each report contains information from one or more source tables. The report descriptions in this topic list source tables. Refer to these source tables if you are interested in learning where certain data originates.

Attributes

Each report has certain attributes associated with it. Use these attributes as *keywords* to search for specific reports in the dialogs.

You can specify any number of attributes for a report, but the area to which the report belongs (for example, UNIX) is always present for predefined reports.

You can also specify these attributes, when appropriate:

- Resource types, such as storage or processor
- Performance issues, such as availability or response
- Presentation forms, such as detail, overview, or trend
- Time resolutions, such as hourly, daily, or monthly

Variables

Each report has variables associated with it. You specify the values for these variables when you generate the report using the reporting dialog.

When you specify a date for a monthly report, specify the first day of the month. Otherwise, there is no match in the data table.

If a character variable happens to have only numeric characters, enclose it in single quote marks, otherwise it will not match the data. For example, if you have a system ID of 1234, specify it as '1234' on the Variables window.

Reports in the accounting subcomponent

This section describes the accounting subcomponent reports:

- "UNIX Acct Commands by User, Daily Overview."
- "UNIX Acct Users by Command, Daily Overview" on page 154.
- "UNIX Acct Cmd Resource Consumption, Daily Overview" on page 155.
- "UNIX Acct User Resource Usage, Monthly Overview" on page 157.
- "UNIX Acct Disk Blocks by User, Monthly Overview" on page 158.
- "UNIX Acct Disk Blocks in 1000s, Monthly Trend" on page 159.
- "UNIX Acct Users and Connects, Daily Overview" on page 160.
- "UNIX Acct Printed Pages by User, Monthly Overview" on page 161.
- "UNIX Acct Printed Pages by System, Monthly Overview" on page 162.

UNIX Acct Commands by User, Daily Overview

For a specific node in the network, this report (see Figure 46 on page 153) provides daily overview information about the resource consumption by user name and

command name. The report is produced by period name (for example, PRIME or NIGHT). You can use the report to identify and have control of, the amount of total resources that each user is consuming.

You can easily amend the information provided in the report to suit your own requirements. For example, you can add the System time and User time values, to have a measure of total processor resource.

The report can be used as a guide for charging users and departments for the system resources they have used.

This information identifies the report:

Report ID
XACCT01

Report group
Accounting Subcomponent Reports

Source
XACCT_COMMAND_D, (described on page "XACCT_COMMAND_D, _M" on page 134)

Attributes
UNIX, Acct, Accounting, Command, Cmd, User, Daily, Overview

Variables
Date, Node name, Period name

UNIX Acct Commands by User, Daily Overview
Date: 2003-12-15 Period: 'PRIME '
Node name: 'sun222 '

User name	Command name	Commands (count)	Elapsed time (hour)	System time (sec)	User time (sec)	Cmd mem avg (KB)	IO (KB)	RW blocks (count)
smith	aixterm	9	0.02	1	3	0.23	7	3
	awk	33	0.09	5	5	0.55	14	5
	.	.	.	.	.	.	.	.
	.	.	.	.	.	.	.	.
	xset	14	0.10	4	10	0.78	18	9
	Total	314	2.19	58	55	5.15	92	80
root	awk	18	0.80	7	5	0.15	13	18
	bsh	38	0.21	8	11	0.89	22	68
	.	.	.	.	.	.	.	.
	.	.	.	.	.	.	.	.
	xmservd	12	1.17	21	24	1.19	51	10
	Total	104	10.18	112	122	5.08	103	130
	Total	418	12.37	170	177	10.23	195	210

Tivoli Decision Support for z/OS Report: XACCT01

Figure 46. Example of UNIX Acct Commands by User, Daily Overview

The report contains this information:

User name
The name of the user who has used this node.

Command name
The name of the UNIX command entered by the user.

Commands (count)
The number of times this user has entered this command.

Elapsed time (hour)
The elapsed time (in hours) used by this user, entering this command. This is calculated as ELAPSED_SEC/3600.

System time (sec)
The system time (in seconds) used by this user, entering this command.

User time (sec)
The user time (in seconds) used by this user, entering this command.

Cmd mem avg (KB)
The average memory (in kilobytes) used by this user, entering this command. This is calculated as MEMORY_BYTES/(1024*COMMANDS).

IO (KB)
Characters (in kilobytes) that have been transferred by this user, using this command. This is calculated as IO_CHARS/1024.

RW blocks (count)
The number of blocks that have been read/written by this user, using this command.

UNIX Acct Users by Command, Daily Overview

For a specific node in the network, this report (see Figure 47 on page 155) provides daily overview information about command usage by each user; the number of times a user has used a command, how much system and user time the command required, how much memory, and how much I/O. The report is produced by period name (for example, PRIME or NIGHT).

You can use the report to identify and have control of, the amount of resources that each user is consuming, by command. The information provided in the report can be amended to suit your own requirements. For example, by modifying the SQL routine that produces the report, you can remove from the report users whose command usage is not relevant.

The report can be used as a guide for charging users and departments for the system resources they have used.

This information identifies the report:

Report ID
XACCT02

Report group
Accounting Subcomponent Reports

Source
XACCT_COMMAND_D, (described on page "XACCT_COMMAND_D, _M" on page 134)

Attributes
UNIX, Acct, Accounting, Command, Cmd, User, Daily, Overview

Variables
Date, Node name, Period name

```
                UNIX Acct Users by Command, Daily Overview
                     Date: 2003-12-16  Period: 'PRIME '
                           Node name: 'hp555 '

                          |          Elapsed   System    User
Command  User             | Commands   time      time     time  Cmd mem       IO  RW blocks
 name    name             | (count)   (hour)    (sec)     (sec) avg (KB)     (KB)  (count)
-------- --------         | -------- -------- -------- -------- -------- -------- ---------
aixterm  smith            |        9     0.05        2        3     0.23       11         2
awk      smith            |        7     0.07        3        2     0.07        2         4
         root             |       18     0.08        2        1     0.15       13         7
bind     smith            |       28     0.10        6        4     0.44       10         3
.        .                |        .        .        .        .        .        .         .
.        .                |        .        .        .        .        .        .         .

                  Tivoli Decision Support for z/OS Report: XACCT02
```

Figure 47. Example of UNIX Acct Users by Command, Daily Overview

The report contains this information:

Command name
The name of the UNIX command.

User name
The name of the user who has used this node.

Commands (count)
The number of times this user has entered this command.

Elapsed time (hour)
The elapsed time (in hours) used by this user, entering this command. This is calculated as ELAPSED_SEC/3600.

System time (sec)
The system time (in seconds) used by this user, entering this command.

User time (sec)
The user time (in seconds) used by this user, entering this command.

Cmd mem avg (KB)
The memory (in kilobytes) used by this user, entering this command. This is calculated as MEMORY_BYTES/(1024*COMMANDS).

IO (KB)
Characters (in kilobytes) that have been transferred by this user, using this command. This is calculated as IO_CHARS/1024.

RW blocks (count)
The number of blocks that have been read/written by this user, using this command.

UNIX Acct Cmd Resource Consumption, Daily Overview

For a specific node in the network, this report (see Figure 48 on page 156) provides daily overview information about command usage; the number of times a command has been used, how much system and user time the command required, how much memory, and how much I/O. The report is produced by period name (for example, PRIME or NIGHT). You can use the report to identify and have control of, the amount of total resources that each command is consuming.

You can easily amend the information provided in the report to suit your own requirements. For example, by modifying the SQL routine that produces the report, you can remove from the report commands that are not particularly important. Or you can compare the amount of resources (for example, System time + User time) a command requires for different nodes. If command `xyz` requires more System

and User time at 'nodeA' than at 'nodeB', you might investigate, for example, if paging is too high at the first node, or if the command has to process more data at the first node.

This information identifies the report:

Report ID
XACCT03

Report group
Accounting Subcomponent Reports

Source
XACCT_COMMAND_D, (described on page "XACCT_COMMAND_D, _M" on page 134)

Attributes
UNIX, Acct, Accounting, Command, Cmd, User, Daily, Overview

Variables
Date, Node name, Period name

UNIX Acct Cmd Resource Consumption, Daily Overview
Date: 2003-12-17 Period: 'PRIME '
Node name: 'aix111 '

Command name	Commands (count)	Elapsed avg (min)	System avg (sec)	User avg (sec)	Cmd mem avg (KB)	IO avg (KB)	RW avg (block)
aixterm	7	0.08	0.23	0.33	0.23	17.0	5
awk	5	0.03	0.04	0.05	0.13	4.1	2
bind	8	0.02	0.09	0.64	0.44	12.5	3
bsh	26	0.02	0.10	0.03	0.07	4.4	1
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.

Tivoli Decision Support for z/OS Report: XACCT03

Figure 48. Example of UNIX Acct Cmd Resource Consumption, Daily Overview

The report contains this information:

Command name
The name of the UNIX command.

Commands (count)
The number of times this command has been entered.

Elapsed avg (min)
The average elapsed time (in minutes) used by this command, This is calculated as ELAPSED_SEC/(COMMANDS*60).

System avg (sec)
The average system time (in seconds) used by this command. This is calculated as SYSTEM_SEC/COMMANDS.

User avg (sec)
The average user time (in seconds) used by this command. This is calculated as USER_SEC/COMMANDS.

Cmd mem avg (KB)
The average memory (in kilobytes) used by this command. This is calculated as MEMORY_BYTES/(1024*COMMANDS).

IO avg (KB)
The average number of characters (in kilobytes) used by this command. This is calculated as (IO_CHARS/1024)/COMMANDS.

RW avg (block)
The average number of blocks read/written by this command. This is calculated as RW_BLOCKS/COMMANDS.

UNIX Acct User Resource Usage, Monthly Overview

For a specific node in the network, this report (see "UNIX Acct User Resource Usage, Monthly Overview") provides monthly overview information about the resource usage by user name. The report would typically be run each month in batch mode, and if the report shows that further investigation for a specific user (or users) is required, you can then use the report "UNIX Acct Commands by User, Daily Overview" on page 152 for a *daily* breakdown of resource usage by the user(s). The report is produced by period name (for example, PRIME or NIGHT).

The report can be used as a guide for charging users and departments for the system resources they have used.

This information identifies the report:

Report ID
XACCT04

Report group
Accounting Subcomponent Reports

Source
XACCT_COMMAND_M, (described on page "XACCT_COMMAND_D, _M" on page 134)

Attributes
UNIX, Acct, Accounting, Command, Cmd, User, Monthly, Overview

Variables
Month, Node name, Period name

```
              UNIX Acct User Resource Usage, Monthly Overview
                     Month: 2003-12-01  Period: 'PRIME '
                             Node name: 'aix555 '

          |            Elapsed   System    User
User      | Commands    time      time      time       IO
name      | (count)    (hour)    (sec)     (sec)     (KB)
----------|-------- -------- -------- -------- --------
smith     |      514     1.19       58      125     3392
root      |      104     0.18        4        2      317

                   Tivoli Decision Support for z/OS Report: XACCT04
```

Figure 49. Example of UNIX Acct User Resource Usage, Monthly Overview

The report contains this information:

User name
The name of the user who has used this node.

Commands (count)
The total number of commands this user has entered.

Elapsed time (hour)
The elapsed time (in hours) used by this user, entering this command. This is calculated as ELAPSED_SEC/3600.

System time (sec)
The system time (in seconds) used by this user, entering this command.

User time (sec)
The user time (in seconds) used by this user, entering this command.

IO (KB)
Characters (in kilobytes) that have been transferred by this user, using this command. This is calculated as IO_CHAR/1024.

UNIX Acct Disk Blocks by User, Monthly Overview

For a specific node in the network, this report (see"UNIX Acct Disk Blocks by User, Monthly Overview") provides monthly overview information about the average disk blocks allocated by each user, *at this node*. The report would typically be run each month in batch mode.

The report can be used as a guide for charging users and departments for the disk resources they have used.

This information identifies the report:

Report ID
XACCT05

Report group
Accounting Subcomponent Reports

Source
XACCT_DISK_M, XACCT_DISK_MV (View), (described on page "XACCT_DISK_MV" on page 137)

Attributes
UNIX, Acct, Accounting, Disk, User, Monthly, Overview

Variables
Month, Node name

```
              UNIX Acct Disk Blocks by User, Monthly Overview
                              Month: 2003-12-01
                            Node name: 'aix555 '

   User     | Disk blocks
   name     |     avg
   ---------|-----------
   adm      |         216
   archer   |           4
   bin      |      398644
   smith    |         188
   daemon   |           4
   guest    |           4
   johanbe  |           4
   loadl    |         116
   root     |      103824
   sys      |           8
   uucp     |         972
   xbill    |         132
            |============
      Total |      504116
                Tivoli Decision Support for z/OS Report: XACCT05
```

Figure 50. Example of UNIX Acct Disk Blocks by User, Monthly Overview

The report contains this information:

User name
The name of the user who has used this node.

Disk blocks avg
The average number of blocks allocated by this user. This is calculated as DISK_BLOCKS/(RECORDS_COLLECTED for USER=root).

UNIX Acct Disk Blocks in 1000s, Monthly Trend

For a specific node in the network, this report (see Figure 51 on page 160) provides monthly trend information about the average disk blocks allocated by the node. The number of blocks is given in units of one thousand. You can use the report to anticipate potential bottlenecks in disk space capacity (where, for example, the usage shows a steadily increasing trend).

The report would typically be run each month in batch mode. If you require more detailed information about a node's disk usage for a *specific month*, you can proceed to the report "UNIX Acct User Resource Usage, Monthly Overview" on page 157.

This information identifies the report:

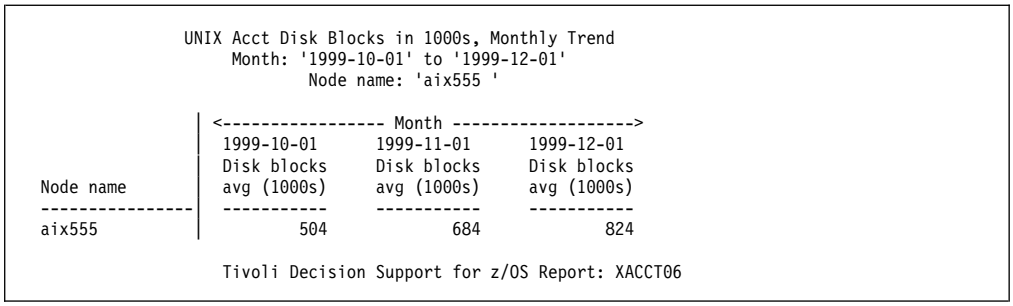
Report ID
XACCT06

Report group
Accounting Subcomponent Reports

Source
XACCT_DISK_M, XACCT_DISK_MV (View), (described on page "XACCT_DISK_MV" on page 137)

Attributes
UNIX, Acct, Accounting, Disk, Monthly, Trend

Variables
From month, To month, Node name



```
                UNIX Acct Disk Blocks in 1000s, Monthly Trend
                     Month: '1999-10-01' to '1999-12-01'
                              Node name: 'aix555 '

                  | <----------------- Month ------------------->
                  |  1999-10-01      1999-11-01      1999-12-01
                  |  Disk blocks     Disk blocks     Disk blocks
Node name         |  avg (1000s)     avg (1000s)     avg (1000s)
----------------- |  -----------     -----------     -----------
aix555            |           504             684             824

                     Tivoli Decision Support for z/OS Report: XACCT06
```

Figure 51. Example of UNIX Acct Disk Blocks in 1000s, Monthly Trend

The report contains this information:

Month
The month start date.

Node name
The name of the node in the network.

Disk blocks avg (1000s)
The average number of blocks (in thousands) used by this node. This is calculated as (DISK_BLOCKS/1000)/(RECORDS_COLLECTED for USER=root).

UNIX Acct Users and Connects, Daily Overview

For a specific node in the network, this report (see Figure 52 on page 161) provides daily overview information about the connects that have been made at the node. The report is produced by period name (for example, PRIME or NIGHT). You can use the report to control the use of the node.

This information identifies the report:

Report ID
XACCT07

Report group
Accounting Subcomponent Reports

Source
XACCT_CONNECT_D, XACCT_CONNECT_TYPE, (described on pages "XACCT_CONNECT_D, _M" on page 136 and "XACCT_CONNECT_TYPE" on page 144 respectively)

Attributes
UNIX, Acct, Accounting, Process, User, Daily, Overview

Variables
Date, Node name, Period name

```
                UNIX Acct Users and Connects, Daily Overview
                      Date: 1999-12-15  Period: 'PRIME '
                             Node name: 'aix555 '

User     |   Connect        Connects
name     | description       (count)
---------| ---------------- ---------
archer   | USER_PROCESS             8
smith    | USER_PROCESS            14
cron     | INIT_PROCESS             1
root     | LOGIN_PROCESS            1

                    Tivoli Decision Support for z/OS Report: XACCT07
```

Figure 52. Example of UNIX Acct Users and Connects, Daily Overview

The report contains this information:

User name
The name of the user who has used this node.

Connect description
The connect type description.

Connects (count)
The total number of connects for this connect type, this user has made.

UNIX Acct Printed Pages by User, Monthly Overview

For a specific node in the network, this report (see Figure 53 on page 162) provides monthly overview information about how much of the printing resources each user has used. The report is produced by period name (for example, PRIME or NIGHT). The report can be used as a guide for charging users and departments for the printing resources they have used.

This information identifies the report:

Report ID
XACCT08

Report group
Accounting Subcomponent Reports

Source
XACCT_PRINT_M, (described on page "XACCT_PRINT_D, _M" on page 136)

Attributes
UNIX, Acct, Accounting, Print, Page, User, Monthly, Overview

Variables
Month, Node name, Period name

```
            UNIX Acct Printed Pages by User, Monthly Overview
                  Month: 1999-01-01  Period: 'PRIME'
                           Node name: 'aix555 '

User     Print queue Request
name        name      node                              Printouts    Pages
-------- ----------- -------------------------------- --------- --------
smith    prt0        aix555                                   2        2
         wn5prt      aix555                                   2        2
                                                      --------- --------
                                                Total         4        4

                                                      ========= ========
                                                Total         4        4

                  Tivoli Decision Support for z/OS Report: XACCT08
```

Figure 53. Example of UNIX Acct Printed Pages by User, Monthly Overview

The report contains this information:

User name
The name of the user who has used this node.

Print queue name
The name of the print queue used by this user.

Request node
The name of the node from which print was requested. When the request node is the same as the node name for which the report is being produced, the print request is local.

Printouts
The number of printouts printed by this user, at this node.

Pages Total number of pages printed by this user, at this node.

UNIX Acct Printed Pages by System, Monthly Overview

For a specific node in the network, this report (see "UNIX Acct Printed Pages by System, Monthly Overview") provides monthly overview information about how much of the printing resources each node has used. The report is produced by period name (for example, PRIME or NIGHT). The report can be used to control the amount of printing being performed by each node.

This information identifies the report:

Report ID
XACCT09

Report group
Accounting Subcomponent Reports

Source
XACCT_PRINT_M, (described on page "XACCT_PRINT_D, _M" on page 136)

Attributes
UNIX, Acct, Accounting, Print, Page, System, Monthly, Overview

Variables
Month, Period name

```
            UNIX Acct Printed Pages by System, Monthly Overview
                   Month: 1999-01-01  Period: 'PRIME'

                  Print queue Request
Node name            name      node    Printouts    Pages
---------------- ----------- -------- --------- --------
sun555           prt0        aix222           2       12
                 wn5prt      aix555           2       52
                                      --------- --------
                                 Total         4       64

                                      ========= ========
                                 Total         4       64

                  Tivoli Decision Support for z/OS Report: XACCT09
```

Figure 54. Example of UNIX Acct Printed Pages by System, Monthly Overview

The report contains this information:

Node name
The name of the node in the network.

Print queue name
The print queue name.

Request node
The node from which the print was requested. When the request node is the same as the node name for which the report is being produced, the print request is local.

Printouts
For the specified request node, the total number of printouts printed.

Pages For the specified request node, the total number of pages printed.

Reports in the configuration subcomponent

This section describes the configuration subcomponent reports:

- "UNIX Configuration of HW for a System, Overview."
- "UNIX Configuration of HW for Device Class, Overview" on page 164.
- "UNIX Configuration of SW for a System, Overview" on page 165.
- "UNIX Configuration of SW for Object, Overview" on page 166.

UNIX Configuration of HW for a System, Overview

This report only covers nodes that use AIX. For a specific node in the network and for a time period determined by the From and To-dates, this report (see Figure 55 on page 164) provides overview information about the devices that were found when the last "collect" job was run, information about each device (status, sub-class, type, date and time of configuration) are given.

This information identifies the report:

Report ID
XCONFIG01

Report group
Configuration Subcomponent Reports

Source
XCONFIG_HW, (described on page "XCONFIG_HW" on page 137)

Attributes
AIX, Configuration, HW, Hardware, Overview

Variables
From date, To date, Node name

UNIX Configuration of HW for a System, Overview
Date: '2003-11-13' to '2003-12-13'
Node name: 'aix555 '

Device class	Device name	Status	Device subclass	Device type	Date	Time
adapter	ent0	A	sio	ient_2	2003-12-13	16.45.12
	fda0	A	sio	fda_2	2003-12-13	16.45.12
	ppa0	A	sio	ppa	2003-12-13	16.45.12
	.	.	.	.	.	.
	.	.	.	.	.	.
	.	.	.	.	.	.
aio	aio0	D	Asynchronous	node	2003-12-13	16.45.12
bus	bus0	A	sys	mca	2003-12-13	16.45.12
	sgabus0	A	sys	sgabus	2003-12-13	16.45.12
disk	hdisk0	A	scsi	1000mb	2003-12-13	16.45.12
diskette	fd0	A	siofd	fd	2003-12-13	16.45.12
en	en0	D	Standard	EN	2003-12-13	16.45.12
fpa	fpa0	A	sys	fpa1	2003-12-13	16.45.12
	.	.	.	.	.	.
	.	.	.	.	.	.
	.	.	.	.	.	.

Tivoli Decision Support for z/OS Report: XCONFIG01

Figure 55. Example of UNIX Configuration of HW for a System, Overview

The report contains this information:

Device class
A class of devices configured at this node.

Device name
The name of a device contained within the given device class.

Status The status of the device contained within the given device class. The possible values are:
A = Active
D = Defined

Device subclass
The subclass of the device contained within the device class.

Device type
The type of the device contained within the device class.

Date The date when the collect was run, and when the configuration data for this device was captured.

Time The time when the collect was run, and when the configuration data for this device was captured.

UNIX Configuration of HW for Device Class, Overview

This report only covers nodes that use AIX. For a specific device class or for all device classes, this report (see Figure 56 on page 165) provides overview information about the number of devices in the network, found when the collect job(s) was run. If no device class is entered, the number of devices for *all* device classes in the network, will be displayed.

The information is displayed for a time period determined by From and To dates that you enter.

This information identifies the report:

Report ID
XCONFIG02

Report group
Configuration Subcomponent Reports

Source
XCONFIG_HW, (described on page "XCONFIG_HW" on page 137)

Attributes
AIX, Configuration, HW, Hardware, Device, Overview

Variables
From date, To date, Device class

```
              UNIX Configuration of HW for Device Class, Overview
                      Date: '2003-11-13' to '2003-12-13'
                           Device class: 'adapter '

                    <----------- Date --------->
                     2003-11-13      2003-12-13
Node name               Devices         Devices
----------------  -----------     -----------
aix555                       12              15

                  Tivoli Decision Support for z/OS Report: XCONFIG02
```

Figure 56. Example of UNIX Configuration of HW for Device Class, Overview

The report contains this information:

Date The date(s) when the collect job was run.

Node name
The name of the node in the network.

Devices
The number of devices that were found by the collect job, on the date(s) the collect was run.

UNIX Configuration of SW for a System, Overview

This report covers only nodes that use AIX. For a specific node in the network, this report (see Figure 57 on page 166) provides overview information about the software configurations that have taken place. The information is displayed for a time period determined by the From and To dates that you enter.

This information identifies the report:

Report ID
XCONFIG03

Report group
Configuration Subcomponent Reports

Source
XCONFIG_SW, (described on page "XCONFIG_SW" on page 137)

Attributes
AIX, Configuration, SW, Software, System, Overview

Variables
From date, To date, Node name

UNIX Configuration of SW for a System, Overview
Date: '2003-01-01' to '2003-07-0 1'
Node name: 'aix555 '

Feature code	Path name	Software object	Install date	Description	Date	Time
	/usr/lib/objrepo	xmconsole.data	09/13/03	XMconsole configuration files	2003-01-17	09.51.31
		xmconsole.obj	09/13/03	XMconsole executables	2003-01-17	09.51.31
	/etc/objrepo	xlccmp.obj	09/10/03	AIX XL C Compiler/6000	2003-01-17	09.51.31
	/usr/lib/objrepo	xlccmp.obj	09/10/03	AIX XL C Compiler/6000	2003-01-17	09.51.31
2924	/usr/lib/objrepo	netmgrmEn_US.msg	09/10/03	Network Management I Messages	2003-01-17	09.51.31
5005	/etc/objrepo	netmgr.am.obj	09/10/03	Alert Manager	2003-01-17	09.51.31
		netmgr.api.obj	09/10/03	SNMP Appl. Programming Interf	2003-01-17	09.51.31
		netmgr.clm.obj	09/10/03	SNMP Command Line Manager	2003-01-17	09.51.31
		netmgr.nvdm.obj	09/10/03	NetView Dist. Manager Catcher	2003-01-17	09.51.31
	/etc/objrepo	nv6000.features.ob	10/29/03	AIX SystemView NetView/6000 Feat	2003-01-17	09.51.31
	/usr/lib/objrepo	nv6000.base.obj	10/29/03	AIX SystemView NetView/6000 Base	2003-01-17	09.51.31
		nv6000.features.ob	10/29/03	AIX SystemView NetView/6000 Feat	2003-01-17	09.51.31
0000	/usr/lib/objrepo	perfmgr	09/10/03	Performance Manager	2003-01-17	09.51.31
0000	/etc/objrepo	perfagent	09/10/03	Performance Agent	2003-01-17	09.51.31
	/usr/lib/objrepo	perfagent	09/10/03	Performance Agent	2003-01-17	09.51.31
5005	/usr/lib/objrepo	aic.obj	09/10/03	AIXwindows Interface Composer	2003-01-17	09.51.31
2924	/usr/lib/objrepo	bsmEn_US.msg	09/10/03	Base System Messages-U.S. Engl	2003-01-17	09.51.31
2937	/usr/lib/objrepo	bsmSv_SE.msg	09/10/03	Base System Messages-Swedish	2003-01-17	09.51.31
5005	/etc/objrepo	bos.obj	01/01/90	The Base Operating System	2003-01-17	09.51.31
		bosext1.ecs.obj	09/10/03	Remote Customer Services	2003-01-17	09.51.31
		bosext1.mh.obj	09/10/03	Mail Handler	2003-01-17	09.51.31

Tivoli Decision Support for z/OS Report: XCONFIG03

Figure 57. Example of UNIX Configuration of SW for a System, Overview

The report contains this information:

Feature code
The feature code identification of the software object that is installed.

Path name
The path to where the software object is located.

Software object
The name of the software object.

Install date
The date when the software object was installed.

Description
A description of the software object.

Date The date of the collect run, when the information about the software object was obtained.

Time The time of the collect run, when the information about the software object was obtained.

UNIX Configuration of SW for Object, Overview

This report only covers nodes that use AIX. For a specific software object, this report (see Figure 58 on page 167) provides overview information about the software objects that exist on the network. If no software object is selected for display, the report will provide information about *all* software objects.

The information is displayed for a time period determined by From and To dates that you enter.

This information identifies the report:

Report ID
XCONFIG04

Report group
Configuration Subcomponent Reports

Source
XCONFIG_SW, (described on page "XCONFIG_SW" on page 137)

Attributes
AIX, Configuration, SW, Software, Object, Overview

Variables
From date, To date, Object

```
              UNIX Configuration of SW for Object, Overview
                    Date: '2003-01-01' to '2003-02-10'
                          Object: 'bos.obj '

Software                                                        Install    Install
object                  Node name  Path name         Release     status      date     Date        Time
----------------------------------- ---------------- ---------- ---------- -------- ---------- --------
bos.obj                 aix555     /etc/obj/repo     03.02.0000.0000 COMPLETE   01/01/03  2003-01-31  15.52.57
                                   /usr/lib/objrepo  03.02.0000.0000 COMPLETE   01/01/03  2003-01-31  15.52.57

                Tivoli Decision Support for z/OS Report: XCONFIG04
```

Figure 58. Example of UNIX Configuration of SW for Object, Overview

The report contains this information:

Software object
The name of the software object that has been selected.

Node name
The node within the network, where the software object is located.

Path name
The path(s) to where the software object is located.

Release
The release level of the software object.

Install status
The status of the installation of the software object.

Install date
The date when the software object was installed.

Date The date of the collect run, when the information about the software object was obtained.

Time The time of the collect run, when the information about the software object was obtained.

Reports in the error subcomponent

This section describes the error subcomponent reports:

- "UNIX Error by ID, Daily Overview" on page 168.
- "UNIX Error by Type, Daily Overview" on page 168.
- "UNIX Error by Class, Daily Overview" on page 169.
- "UNIX Error by Resource, Daily Overview" on page 170.
- "UNIX Error by Resource, Monthly Trend" on page 171.

UNIX Error by ID, Daily Overview

This report only covers nodes that use AIX. For a specific node in the network, this report (see Figure 59) gives daily overview information about the error messages that have been issued. For each error message, the report shows error ID, description, number of times the message has occurred, and the percentage occurrence of the message.

You can use this report to identify potential problems within an application, since the error ID allows you to differentiate between errors caused by software applications, system errors, and so on.

This information identifies the report:

Report ID
XERROR01

Report group
Error Subcomponent Reports

Source
XERROR_D, (described on page "XERROR_D, _M" on page 138)

Attributes
AIX, Error, ID, Daily, Overview

Variables
Date, Node name

```
                    UNIX Error by ID, Daily Overview
                            Date: 1999-11-29
                          Node name: 'aix555 '

                                                          Errors     Errors
Error ID | Description                                    (count)     (%)
---------|-----------------------------------------  ----------  --------
0F27AAE5 | SOFTWARE PROGRAM ABNORMALLY TERMINATED             6     54.55
E18E984F | SOFTWARE PROGRAM ERROR                             3     27.27
1104AA28 | System reset interrupt received                    1      9.09
9DBCFDEE | Error logging turned on                            1      9.09
         |                                                ==========  ========
         |                                       Total          11    100.00

                Tivoli Decision Support for z/OS Report: XERROR01
```

Figure 59. Example of UNIX Error by ID, Daily Overview

The report contains this information:

Error ID
The identification of the error message.

Description
Description of the error message.

Errors (count)
The number of times this error has occurred.

Errors (%)
The percentage of the total errors, in which this message has occurred.

UNIX Error by Type, Daily Overview

This report only covers nodes that use AIX. For a specific node in the network, this report (see Figure 60 on page 169) gives daily overview information about the

types of error messages that have been issued. For each error type, the report shows the number of times the message type has occurred, and the percentage occurrence of the message type.

You can use this report to identify potential problems within a system, since the message resource type allows you to measure the number of errors that are being generated for a particular error type.

This information identifies the report:

Report ID
XERROR02

Report group
Error Subcomponent Reports

Source
XERROR_D, (described on page "XERROR_D, _M" on page 138)

Attributes
AIX, Error, Type, Daily, Overview

Variables
Date, Node name

```
                    UNIX Error by Type, Daily Overview
                             Date: 1999-11-29
                           Node name: 'aix555 '

             |     Errors     Errors
Error type   |     (count)     (%)
-------------|  ----------  --------
P            |           9     81.82
T            |           2     18.18
             |  ==========  ========
       Total |          11    100.00

                  Tivoli Decision Support for z/OS Report: XERROR02
```

Figure 60. Example of UNIX Error by Type, Daily Overview

The report contains this information:

Error type
The error type, which can be the following:
P = Program error
T = Token ring error

Errors (count)
The number of times this error type has occurred.

Errors (%)
The percentage of the total errors, that this error type has occurred.

UNIX Error by Class, Daily Overview

This report only covers nodes that use AIX. For a specific node in the network, this report (see Figure 61 on page 170) gives daily overview information about the class of error messages that have been issued. For each error class, the report shows the number of times the message has occurred, and the percentage occurrence of the message class.

You can use this report to identify potential problems within a system, since the message resource type allows you to measure the number of errors that are being generated for a particular error class.

This information identifies the report:

Report ID
XERROR03

Report group
Error Subcomponent Reports

Source
XERROR_D, (described on page "XERROR_D, _M" on page 138)

Attributes
AIX, Error, Class, Daily, Overview

Variables
Date, Node name

```
                  UNIX Error by Class, Daily Overview
                           Date: 1999-11-29
                         Node name: 'aix555 '

                 Errors     Errors
Error class      (count)      (%)
-----------  ----------  --------
S                    10     90.91
O                     1      9.09
             ==========  ========
      Total          11    100.00

                Tivoli Decision Support for z/OS Report: XERROR03
```

Figure 61. Example of UNIX Error by Class, Daily Overview

The report contains this information:

Error class
The error class, which can be the following:
S = Software message
H = Hardware message
O = Error log command message

Errors (count)
The number of times this error class has occurred.

Errors (%)
The percentage of the total errors, that this error class has occurred.

UNIX Error by Resource, Daily Overview

This report only covers nodes that use AIX. For a specific node in the network, this report (see Figure 62 on page 171) gives daily overview information about the resource type of error messages that have been issued. For each resource type, the report shows the number of times messages of this type have occurred, and the percentage occurrence of messages of this resource type.

You can use this report to identify potential problems within a system, since the message resource type allows you to measure the number of errors that are being generated for a particular resource.

This information identifies the report:

Report ID
XERROR04

Report group
Error Subcomponent Reports

Source
XERROR_D, (described on page "XERROR_D, _M" on page 138)

Attributes
AIX, Error, Resource, Daily, Overview

Variables
Date, Node name

```
                UNIX Error by Resource, Daily Overview
                           Date: 1999-11-29
                         Node name: 'aix555 '

Resource           Errors       Errors
  name             (count)       (%)
--------------  ----------  ----------
SYSPROC                  7       63.64
SRC                      3       27.27
errdemon                 1        9.09
              ==========  ==========
         Total          11      100.00

                Tivoli Decision Support for z/OS Report: XERROR04
```

Figure 62. Example of UNIX Error by Resource, Daily Overview

The report contains this information:

Resource name
The name of the resource.

Errors (count)
The number of times errors originating from this resource name, have occurred.

Errors (%)
The percentage of the total errors, that have originated from this resource.

UNIX Error by Resource, Monthly Trend

This report only covers nodes that use AIX. For a specific node in the network, this report (see Figure 63 on page 172) gives monthly trend information about the resource name of error messages that have been issued. For each resource name, the report shows the number of times messages of the resource name have occurred.

You can use this report to identify potential problems within a system, since the message resource type allows you to measure the number of errors that are being generated for a particular resource.

This information identifies the report:

Report ID
XERROR05

Report group
Error Subcomponent Reports

Source
XERROR_M, (described on page "XERROR_D, _M" on page 138)

Attributes
AIX, Error, Resource, Monthly, Trend

Variables
From month, To month, Node name

```
                  UNIX Error by Resource, Monthly Trend
                   Month: '2003-11-01' to '2003-12-01'
                           Node name: 'aix555 '

               <-------- Month -------->
                2003-11-01   2003-12-01
Resource name        Errors       Errors
--------------  -----------  -----------
errdemon                  1            1
tok0                      1            1
SRC                       3            4
SYSPROC                  19           18
SYSVMM                    3            5

                 Tivoli Decision Support for z/OS Report: XERROR05
```

Figure 63. Example of UNIX Error by Resource, Monthly Trend

The report contains this information:

Month
The start date of the month.

Resource name
The name of the resource.

Errors The number of times errors originating from this resource name, have occurred.

Reports in the performance subcomponent

This section describes the performance subcomponent reports:

UNIX Perf CPU Utilization by System, Hourly Trend

For a specific node in the network, this graphical representation (see Figure 64) shows the hourly trend of processor utilization, over a specified time period. Such information is useful as an entry point when investigating system performance.

This information identifies the display:

Report ID
XPERF01

Report group
Performance Subcomponent Reports

Source
XPERF_CPU_H, (described on page "XPERF_CPU_H, _D, _M" on page 139)

Attributes
UNIX, Performance, CPU, Utilization, Usage, Hourly, Trend

Variables
Date, Node name

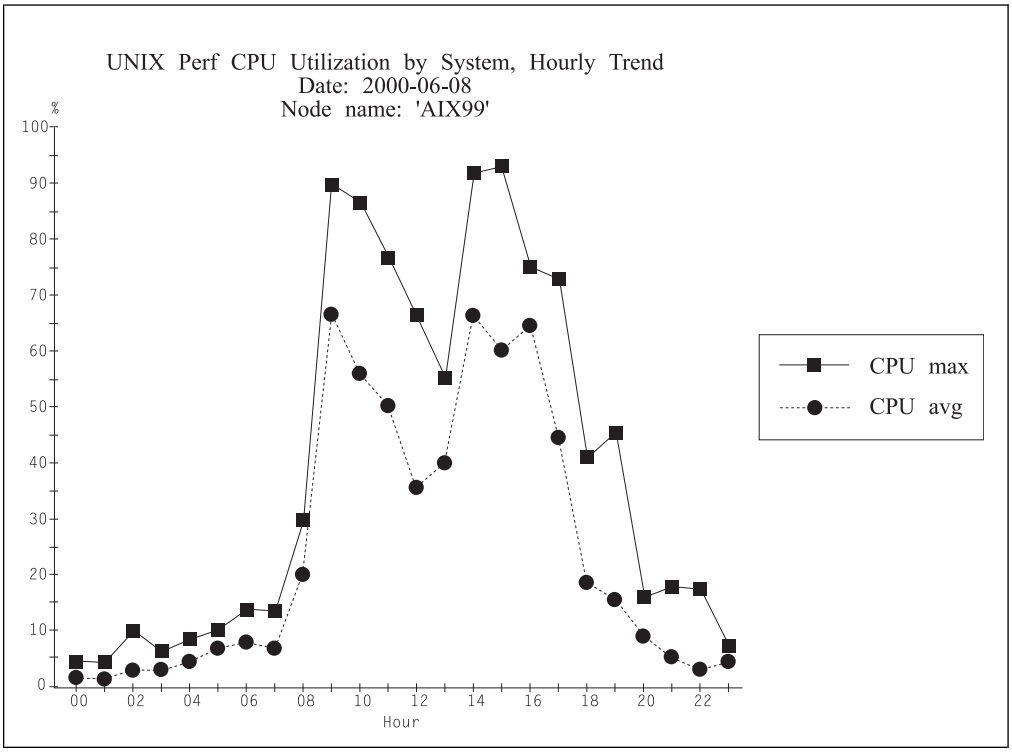


Figure 64. Example of UNIX Perf CPU Utilization by System, Hourly Trend

The report contains this information:

Hour The hour of the day.

CPU avg
The average processor time that has been utilized, in percent.

CPU max
The maximum processor time that has been utilized, in percent.

UNIX Perf CPU Utilization by System, Daily Overview

For all nodes in the network, this graphical representation (see Figure 65) shows the average daily processor utilization. The display is produced by period name (for example, PRIME or NIGHT). Such information is useful as an entry point when investigating system performance. An hourly graphical display of processor utilization for a *specific node*, is given in "UNIX Perf CPU Utilization by System, Hourly Trend" on page 173.

This information identifies the report:

Report ID
XPERF02

Report group
Performance Subcomponent Reports

Source
XPERF_CPU_D, (described on page "XPERF_CPU_H, _D, _M" on page 139)

Attributes
UNIX, Performance, Cpu, Utilization, Usage, Daily, Overview

Variables
Date, Period name

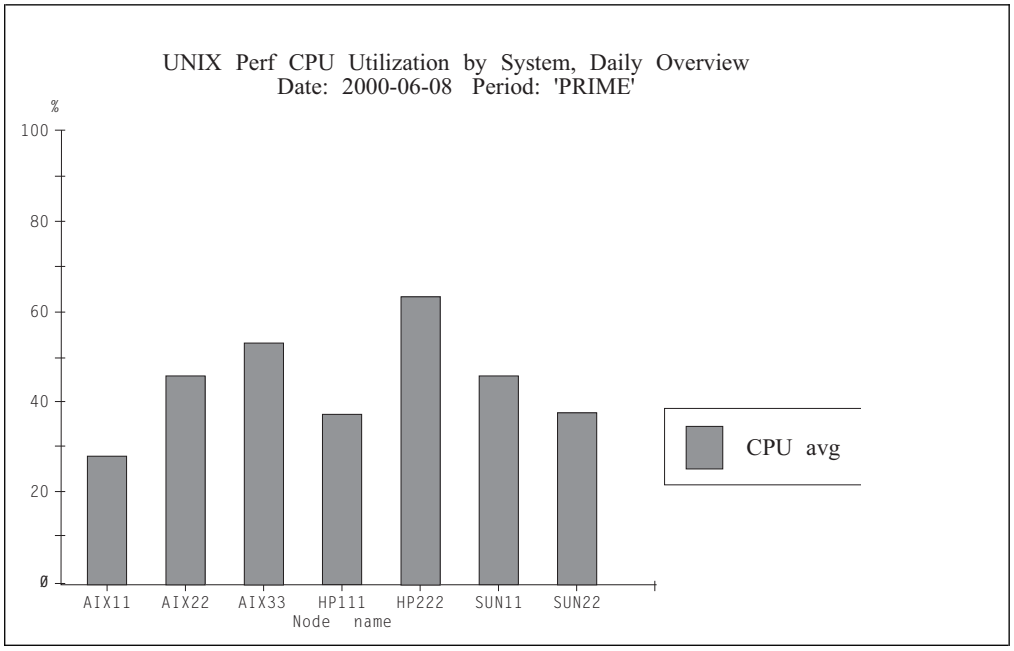


Figure 65. Example of UNIX Perf CPU Utilization by System, Daily Overview

The report contains this information:

Node name
The name of the node in the network.

CPU avg
The average processor time that has been utilized, in percent.

UNIX Perf Statistics by System, Hourly Trend

For a specific node in the network, this report (see Figure 66) provides hourly, trend information about:

- Processor utilization (average and maximum)
- The number of available pages of memory (average and minimum)
- The paging rate (average and maximum)
- The number of processes (average and maximum)
- The number of users (average and maximum)

A total row is given, which are the averages and maximums calculated for all hours.

This information identifies the report:

Report ID
XPERF03

Report group
Performance Subcomponent Reports

Source
XPERF_CPU_H, (described on page "XPERF_CPU_H, _D, _M" on page 139)

Attributes
UNIX, Performance, System, Usage, Hourly, Trend

Variables
Date, Node name

UNIX Perf Statistics by System, Hourly Trend
Date: 1999-12-27
Node name: 'hp111 '

Hour	CPU usage avg (%)	CPU usage max (%)	Mem free pages avg	Mem free pages min	Paging rate avg	Paging rate max	Processes avg	Processes max	Users avg	Users max
.	.	.	.	.	.	.	.	.	.	.
14	6.7	43	1099	166	0.03	0.17	90.3	92	7.0	7
15	4.8	50	1257	687	0.05	0.12	91.3	96	7.3	12
16	5.6	21	822	125	0.02	0.09	92.3	99	7.0	7
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
Total	4.7	50	2447	125	0.06	0.17	85.8	99	6.2	12

Tivoli Decision Support for z/OS Report: XPERF03

Figure 66. Example of UNIX Perf Statistics by System, Hourly Trend2

The report contains this information:

Hour The hour of the day.

CPU usage avg (%)
The average processor utilization, in percent.

CPU usage max (%)
The maximum processor utilization, in percent.

Mem free pages avg
The average number of free pages of memory.

Mem free pages min
The minimum number of free pages of memory.

Paging rate avg
The average paging rate, per second.

Paging rate max
The maximum paging rate, per second.

Processes avg
The average number of processes.

Processes max
The maximum number of processes.

Users avg
The average number of users for this node.

Users max
The maximum number of users for this node.

UNIX Perf Statistics all Systems, Daily Overview

For all nodes in the network, this report (see Figure 67 on page 177) provides daily overview information about:

- Processor utilization (average and maximum)
- The number of available pages of memory (average and minimum)
- The paging rate (average and maximum)
- The number of processes (average and maximum)
- The number of users (average and maximum)

The report is produced by period name (for example, PRIME or NIGHT). An hourly breakdown of the information provided in this report, and for a specific node, is given in "UNIX Perf Statistics by System, Hourly Trend" on page 175.

This information identifies the report:

Report ID
XPERF04

Report group
Performance Subcomponent Reports

Source
XPERF_CPU_D, (described on page "XPERF_CPU_H, _D, _M" on page 139)

Attributes
UNIX, Performance, System, Usage, Daily, Overview

Variables
Date, Period name

UNIX Perf Statistics all Systems, Daily Overview
Date: 1999-12-27 Period: 'PRIME '

Node name	CPU usage avg (%)	CPU usage max /%)	Mem free pages avg	Mem free pages min	Paging rate avg	Paging rate max	Processes avg	Processes max	Users avg	Users max
hp111	5.6	50	1068	125	0.01	2	91.4	96	7.1	8
sun333	18.2	74	621	62	0.15	18	163.9	202	13.5	17

Tivoli Decision Support for z/OS Report: XPERF04

Figure 67. Example of UNIX Perf Statistics all Systems, Daily Overview

The report contains this information:

Node name
The name of the node in the network.

CPU usage avg (%)
The average processor utilization, in percent.

CPU usage max (%)
The maximum processor utilization, in percent.

Mem free pages avg
The average number of free pages of memory.

Mem free pages min
The minimum number of free pages of memory.

Paging rate avg
The average paging rate, per second.

Paging rate max
The maximum paging rate, per second.

Processes avg
The average number of processes.

Processes max
The maximum number of processes.

Users avg
The average number of users for this node.

Users max
The maximum number of users for this node.

UNIX Perf Vol Group and File Syst, Daily Overview

This report only covers nodes that use AIX. For a specific node in the network, this report (see Figure 68 on page 178) provides daily overview information about space utilization: average allocated and free space (in megabytes), and the average percentage used space (by space name and device name). The report is produced by period name (for example, PRIME or NIGHT).

This information identifies the report:

Report ID
XPERF05

Report group
Performance Subcomponent Reports

Source
XPERF_DISK_D, (described on page "XPERF_DISK_D, _M" on page 140)

Attributes
AIX, Performance, Volume, File, Utilization, Usage, Daily, Overview

Variables
Date, Node name, Period name

```
            UNIX Perf Vol Group and File syst, Daily Overview
                    Date: 1999-12-27  Period: 'PRIME '
                            Node name: 'aix555'

                     File          Space  Space size Space free Space used
Volume              system          type   avg (MB)   avg (MB)   avg (%)
------------------ --------------- -----  ---------- ---------- ----------
/                  /dev/hd4        FS          8.0        0.0       92.0
/tmp               /dev/hd3        FS         12.0       11.0        6.0
/usr               /dev/hd2        FS        556.0       67.0       87.0
/usr/lpp/ileaf6.   /dev/lv01       FS        196.0       38.0       80.0
/var               /dev/hd9var     FS         16.0        5.3       65.0
rootvg                             VG       2212.0     1160.0       47.6

                  Tivoli Decision Support for z/OS Report: XPERF05
```

Figure 68. Example of UNIX Perf Vol Group and File Syst, Daily Overview

The report contains this information:

Volume
The volume used by the node.

File system
The file system corresponding to the given volume.

Space type
The space type, which can be one of the following:
FS = File system
VG = Volume group

Space size avg (MB)
The average size of the space, in megabytes. This is calculated as SPACE_SIZE_MB/RECORDS_COLLECTED.

Space free avg (MB)
The average size of free space, in megabytes. This is calculated as SPACE_FREE_MB/RECORDS_COLLECTED.

Space used avg (%)
The average size of used space, in percent. This is calculated as SPACE_USED_PCT/RECORDS_COLLECTED.

UNIX Perf Disk I/O for a Disk, Hourly Trend

For a specific node in the network and a specific disk, this report (see Figure 69 on page 179) provides hourly trend information about the disk's utilization: the average and maximum percentage busy times, the average and maximum read amounts (in kilobytes), and the average and maximum write amounts (in kilobytes).

This information identifies the report:

Report ID
XPERF06

Report group
Performance Subcomponent Reports

Source
XPERF_DISKIO_H, (described on page "XPERF_DISKIO_H, _D, _M" on page 140)

Attributes
UNIX, Performance, Disk, I/O, System, Hourly, Trend

Variables
Date, Disk, Node name

UNIX Perf Disk I/O for a Disk, Hourly Trend
Date: 1999-12-09
Node name: 'aix555 ' Disk: 'hdisk8'

Hour	Busy avg (%)	Busy max (%)	Read avg (KB/sec)	Read max (KB/sec)	Write avg (KB/sec)	Write max (KB/sec)	RW avg (KB/sec)	RW max (KB/sec)
12	12.9	110.6	1.7	52.0	5.4	18.0	7.1	70.0
13	8.9	40.9	0.4	14.0	4.3	8.0	4.7	22.0
14	8.6	24.2	0.1	6.0	4.4	8.0	4.5	14.0
15	53.1	361.1	20.0	219.0	11.3	67.0	31.3	286.0
16	16.3	87.2	3.7	35.0	5.1	13.0	8.8	48.0
17	17.3	115.3	3.2	55.0	6.2	14.0	9.4	69.0
18	8.1	19.9	0.2	5.0	4.1	6.0	4.3	11.0
19	7.6	10.0	1.0	10.2	3.6	5.0	4.6	15.2
20	7.7	9.7	0.5	4.1	4.9	5.0	5.4	9.1
21	7.7	10.6	1.5	17.3	4.0	4.0	5.5	21.3
22	7.7	9.8	0.1	1.9	2.2	5.0	2.3	6.9
23	7.6	9.8	0.3	2.2	4.5	5.0	4.8	7.2
Avg/max	13.6	361.1	2.7	219.0	5.0	67.0	7.7	286.0

Tivoli Decision Support for z/OS Report: XPERF06

Figure 69. Example of UNIX Perf Disk I/O for a Disk, Hourly Trend

The report contains this information:

Hour The hour of the day.

Busy avg (%)
The average time in which the processor was busy, in percent.

Busy max (%)
The maximum time in which the processor was busy, in percent.

Read avg (KB/sec)
The average amount of disk space read per second, in kilobytes. This is calculated as READ_KB_TOTAL/MEASURED_SEC.

Read max (KB/sec)
The maximum amount of disk space read per second, in kilobytes.

Write avg (KB/sec)
The average amount of disk space written per second, in kilobytes. This is calculated as WRITE_KB_TOTAL/MEASURED_SEC.

Write max (KB/sec)
The maximum amount of disk space written per second, in kilobytes.

RW avg (KB/sec)
The average amount of disk space read and written per second, in kilobytes. This is calculated as RW_KB_TOTAL/MEASURED_SEC.

RW max (KB/sec)
The maximum amount of disk space read and written per second, in kilobytes.

UNIX Perf Disk I/O for System, Daily Overview

For a specific node in the network, this report (see Figure 70) provides daily overview information about the utilization of disks at the node: the average and maximum percentage busy times, the average and maximum read amounts (in kilobytes), and the average and maximum write amounts (in kilobytes). The report is produced by period name (for example, PRIME or NIGHT). If you require more detailed hourly trend information about a disk's utilization, you can proceed to the report "UNIX Perf Disk I/O for a Disk, Hourly Trend" on page 178.

This information identifies the report:

Report ID
XPERF07

Report group
Performance Subcomponent Reports

Source
XPERF_DISKIO_D, (described on page "XPERF_DISKIO_H, _D, _M" on page 140)

Attributes
UNIX, Performance, Disk, I/O, System, Daily, Overview

Variables
Date, Node name, Period name

```
               UNIX Perf Disk I/O for System, Daily Overview
                    Date: 1999-12-27  Period: 'PRIME '
                            Node name: 'hp111'

             Busy avg   Busy max   Read avg   Read max  Write avg  Write max
Disk name      (%)        (%)      (KB/sec)   (KB/sec)  (KB/sec)   (KB/sec)
---------- ---------- ---------- ---------- ---------- ---------- ----------
hdisk0           33.5       17.0        1.0        1.4       11.1       40.0
hdisk1           11.4       14.5        1.1        1.4       10.4       12.2
hdisk2           10.0       10.0        1.0        1.0       10.0        1.2
hdisk3           14.3       19.1        1.5       13.0       11.1       18.8
         ========== ========== ========== ========== ========== ==========
   Avg/max       17.3       19.1        1.1       13.0       10.6       40.0

                 Tivoli Decision Support for z/OS Report: XPERF07
```

Figure 70. Example of UNIX Perf Disk I/O for System, Daily Overview

The report contains this information:

Disk name
The name of the physical disk.

Busy avg (%)
The average time in which the disk was busy, in percent.

Busy max (%)
The maximum time in which the disk was busy, in percent.

Read avg (KB/sec)
The average amount of disk that was read per second, in kilobytes. This is calculated as READ_KB_TOTAL/MEASURED_SEC.

Read max (KB/sec)
The maximum amount of disk that was read per second, in kilobytes.

Write avg (KB/sec)
The average amount of disk that was written per second, in kilobytes. This is calculated as WRITE_KB_TOTAL/MEASURED_SEC.

Write max (KB/sec)
The maximum amount of disk that was written per second, in kilobytes.

UNIX Perf Page Space Utilization, Hourly Trend

For a specific node in the network, this report (see Figure 71 on page 182) provides daily overview information about the utilization of page space at the node. For each page space, information about the disk name for the page space, page space size, and the amount of page space used (in kilobytes and also as a percentage), is given.

This information identifies the report:

Report ID
XPERF08

Report group
Performance Subcomponent Reports

Source
XPERF_PAGING_H, (described on page "XPERF_PAGING_H, _D, _M" on page 141)

Attributes
UNIX, Performance, Page, Space, Utilization, Hourly, Trend

Variables
Date, Node name

```
                UNIX Perf Page Space Utilization, Hourly Trend
                             Date: 1999-01-14
                           Node name: 'hp111'

      Page space             Page space Page space Page space
 Hour    name     Disk name  size (MB)  used (MB)   used (%)
----- ---------- ---------- ---------- ---------- ----------
    . .          .                   .          .          .
    8 hd6        hdisk0             96         63       66.1

    9 hd6        hdisk0             96         65       67.7

   10 hd6        hdisk0             96         66       68.9

   11 hd6        hdisk0             96         67       69.8

   12 hd6        hdisk0             96         67       69.8

   13 hd6        hdisk0             96         67       69.8

   14 hd6        hdisk0             96         68       71.2

   15 hd6        hdisk0             96         68       70.8

   16 hd6        hdisk0             96         67       70.0

   17 hd6        hdisk0             96         68       70.8

   18 hd6        hdisk0             96         67       70.3

   19 hd6        hdisk0             96         68       70.8

   20 hd6        hdisk0             96         68       70.8

                 Tivoli Decision Support for z/OS Report: XPERF08
```

Figure 71. Example of UNIX Perf Page Space Utilization, Hourly Trend

The report contains this information:

Hour The hour of the day.

Page space name
The name of the page space used by the node.

Disk name
The physical disk name.

Page space size (MB)
The size of the page space, in megabytes.

Page space used (MB)
The amount of page space used, in megabytes.

Page space used (%)
The amount of page space used, in percent. This is calculated as 100*PS_USED_AVG_KB/PS_SIZE_AVG_KB.

LINUX Percentile Work Size

This report shows the work space allocation for the disks on the file system.

The following information identifies the report:

Report ID:
XLINU01

Report group:
LINUX Reports

Source:
XDISK_INFOR

Attributes:
PERFORMANCE, CPU, UTILIZATION, USAGE, HOURLY, TREND, UNIX, LINUX

```
                    LINUX - PERCENTILE WORK SIZE


DATE        PERIOD    FILE SYS      MAX SIZE  % WORK SIZE
----------  --------  ---------  -----------  -----------
2001-01-16  PRIME     /dev/hda7     23474136            9
                      /dev/hda1        54416           10
                      /dev/hda5      5044156            6

            NIGHT     /dev/hda7    164318952            9
                      /dev/hda1       380912           10
                      /dev/hda5     35309092            6
```

Figure 72. Example of a LINUX - Percentile Work Size report

The report contains the following information:

DATE Date of the measurement.

PERIOD
Name of the period.

FILE SYS
Name of the file system.

MAX SIZE
Maximum size of the disk.

% WORK SIZE
Percentage work space on the disk.

LINUX Disk Space Allocation

This report shows the free space allocation for the disks on the file system.

The following information identifies the report:

Report ID:
XLINU02

Report group:
LINUX Reports

Source:
XDISK_INFOR

Attributes:
PERFORMANCE, CPU, UTILIZATION, USAGE, HOURLY, TREND, UNIX, LINUX

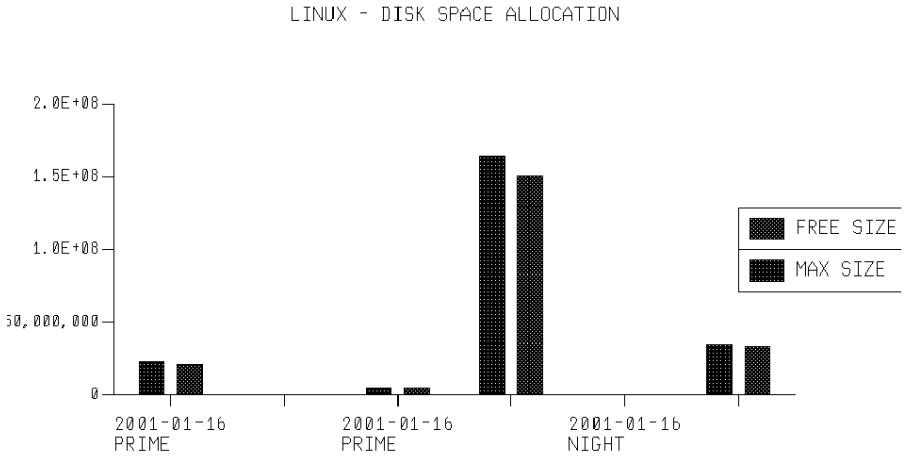


Figure 73. Example of a LINUX - Disk Space Allocation report

The report contains the following information:

DATE Date of the measurement.

PERIOD
Name of the period.

FREE SIZE
Free space on the disk.

MAX SIZE
Maximum size of the disk.

LINUX Performance from User Memory

This report shows 'virtual memory' performance for user memory.

The following information identifies the report:

Report ID:
XLINU03

Report group:
LINUX Reports

Source:
XPERF_VM_INFO

Attributes:
PERFORMANCE, CPU, UTILIZATION, USAGE, HOURLY, TREND, UNIX, LINUX

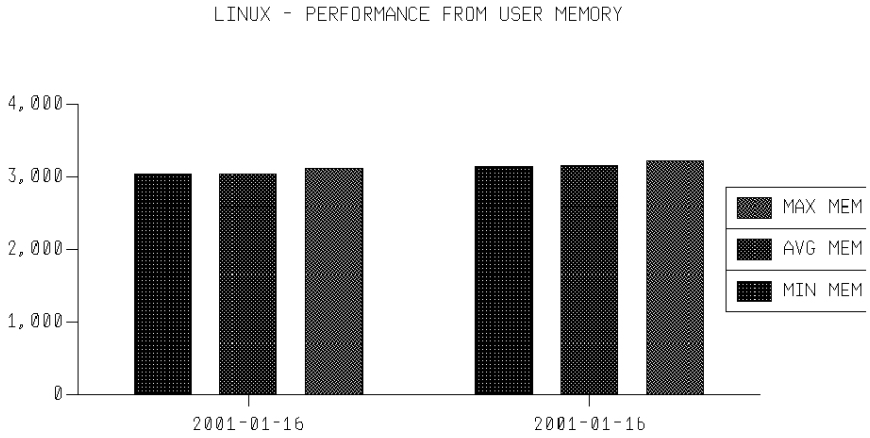


Figure 74. Example of a LINUX - Performance from User Memory report

The report contains the following information:

DATE Date of the measurement.

PERIOD
Name of the period.

MAX MEM
Maximum memory used.

AVG MEM
Average memory used.

MIN MEM
Minimum memory used.

LINUX Performance 'PS' for Volumes Info

This report shows 'physical size' data for the volumes used.

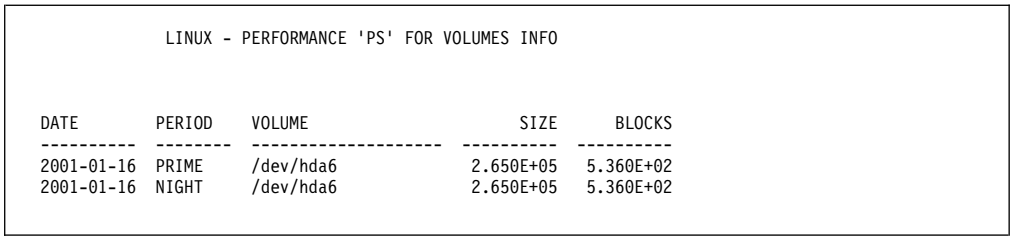
The following information identifies the report:

Report ID:
XLINU04

Report group:
LINUX Reports

Source:
XPERF_PS_INFO

Attributes:
PERFORMANCE, CPU, UTILIZATION, USAGE, HOURLY, TREND, UNIX, LINUX



LINUX - PERFORMANCE 'PS' FOR VOLUMES INFO

DATE	PERIOD	VOLUME	SIZE	BLOCKS
2001-01-16	PRIME	/dev/hda6	2.650E+05	5.360E+02
2001-01-16	NIGHT	/dev/hda6	2.650E+05	5.360E+02

Figure 75. Example of a LINUX - Performance 'PS' for Volumes Info report

The report contains the following information:

DATE Date of the measurement.

PERIOD
Name of the period.

VOLUME
Name of the physical volume.

SIZE Size of the physical volume.

BLOCKS
Number of blocks used on the volume.

LINUX Hardware Configuration

This report shows hardware configuration data for the devices used.

The following information identifies the report:

Report ID:
XLINU05

Report group:
LINUX Reports

Source:
XCONF_HARDWARE

Attributes:
PERFORMANCE, CPU, UTILIZATION, USAGE, HOURLY, TREND, UNIX, LINUX

```
            LINUX - HARDWARE CONFIGURATION


DATE        PERIOD    DEVICE      CPU         ADDRESS     ASSIGNM  BUS

----------  --------  ----------  ----------  ----------  -------  -------
2001-01-16  NIGHT     OTHER       8086        1130                 PCI

2001-01-16  NIGHT     OTHER       8086        1131                 PCI

2001-01-16  NIGHT     OTHER       8086        244e                 PCI

2001-01-16  NIGHT     OTHER       8086        2440                 PCI

2001-01-16  NIGHT     OTHER       8086        244b                 PCI

2001-01-16  NIGHT     OTHER       8086        2442                 PCI

2001-01-16  NIGHT     OTHER       8086        2443                 PCI

2001-01-16  NIGHT     NETWORK     8086        2449        eth      PCI
```

Figure 76. Example of a LINUX - Hardware Configuration report

The report contains the following information:

DATE Date of the measurement.

PERIOD
Name of the period.

DEVICE
Device class of the hardware element.

CPU CPU identification.

ADDRESS
Address ID of the UNIX device.

ASSIGNM
Type of device used in the UNIX system.

BUS Type of bus used in the UNIX system.

LINUX Software Configuration

This report shows software configuration data for the packages used.

The following information identifies the report:

Report ID:
XLINU06

Report group:
LINUX Reports

Source:
XCONF_SOFTWARE

Attributes:
PERFORMANCE, CPU, UTILIZATION, USAGE, HOURLY, TREND, UNIX, LINUX

LINUX - SOFTWARE CONFIGURATION

DATE	PERIOD	PACKAGE	VERSION
2001-01-16	NIGHT	anacron	2.1
2001-01-16	NIGHT	apmd	3.0
2001-01-16	NIGHT	ash	0.2
2001-01-16	NIGHT	at	3.1
2001-01-16	NIGHT	audiofile	0.1
2001-01-16	NIGHT	audiofile-devel	0.1
2001-01-16	NIGHT	aumix	1.3
2001-01-16	NIGHT	authconfig	3.0
2001-01-16	NIGHT	authconf	2.1

Figure 77. Example of a LINUX - Software Configuration report

The report contains the following information:

DATE Date of the measurement.

PERIOD
Name of the period.

PACKAGE
Name of the package used on the system.

VERSION
Software version of the package installed on the system.

LINUX User Information

This report shows process information related to users.

The following information identifies the report:

Report ID:
XLINU07

Report group:
LINUX Reports

Source:
XWTMP_INFOR

Attributes:
PERFORMANCE, CPU, UTILIZATION, USAGE, HOURLY, TREND, UNIX, LINUX

```
                LINUX - USER INFORMATION


DATE        PERIOD    USER        ENTRY  PROCED  NODE

----------  --------  ----------  -----  ------  --------------
2001-01-08  PRIME                 08     000007  2.2.14-12

2001-01-08  PRIME     reboot      02     000000  2.2.14-12

2001-01-08  PRIME     runlevel    01     020019  2.2.14-12

2001-01-08  PRIME     LOGIN       06     000590

2001-01-08  PRIME     root        07     000590

2001-01-08  NIGHT     runlevel    01     013616  2.2.14-12

2001-01-08  NIGHT                 08     000599  2.2.14-12

2001-01-08  NIGHT     shutdown    01     000000  2.2.14-12
```

Figure 78. Example of a LINUX - User Information report

The report contains the following information:

DATE Date of the measurement.

PERIOD
Name of the period.

USER User login name.

ENTRY
Entry type.

PROCED
Process ID.

NODE
Node name.

LINUX Process Information

This report shows process information related to users.

The following information identifies the report:

Report ID:
XLINU08

Report group:
LINUX Reports

Source:
XDTMP_INFOR

Attributes:
PERFORMANCE, CPU, UTILIZATION, USAGE, HOURLY, TREND, UNIX, LINUX

LINUX - PROCESS INFORMATION

DATE	PERIOD	ID	NAME	BLOCKS
2001-01-17	NIGHT	0	root	782380
2001-01-17	NIGHT	1	bin	0
2001-01-17	NIGHT	2	daemon	8
2001-01-17	NIGHT	3	adm	0
2001-01-17	NIGHT	4	lp	0
2001-01-17	NIGHT	5	sync	0
2001-01-17	NIGHT	6	shutdown	0
2001-01-17	NIGHT	7	halt	0

Figure 79. Example of a LINUX - Process Information report

The report contains the following information:

DATE Date of the measurement.

PERIOD
Name of the period.

ID User ID.

NAME
User name.

BLOCKS
Number of disk blocks that are used.

LINUX Performance 'VM' for Swap Memory

This report shows 'virtual memory' performance for swap memory.

The following information identifies the report:

Report ID:
XLINU09

Report group:
LINUX Reports

Source:
XPERF_VM_INFO

Attributes:
PERFORMANCE, CPU, UTILIZATION, USAGE, HOURLY, TREND, UNIX, LINUX

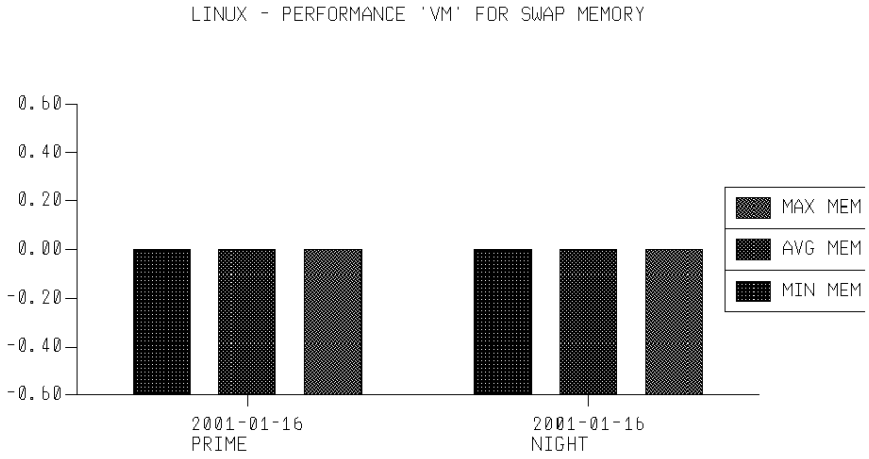


Figure 80. Example of a LINUX - Performance 'VM' for Swap Memory report

The report contains the following information:

DATE Date of the measurement.

PERIOD
Name of the period.

MAX MEM
Maximum swap memory used.

AVG MEM
Average swap memory used.

MIN MEM
Minimum swap memory used.

LINUX Performance 'VM' for CPU

This report shows 'virtual memory' performance for CPU activity.

The following information identifies the report:

Report ID:
XLINU10

Report group:
LINUX Reports

Source:
XPERF_VM_INFO

Attributes:
PERFORMANCE, CPU, UTILIZATION, USAGE, HOURLY, TREND, UNIX, LINUX

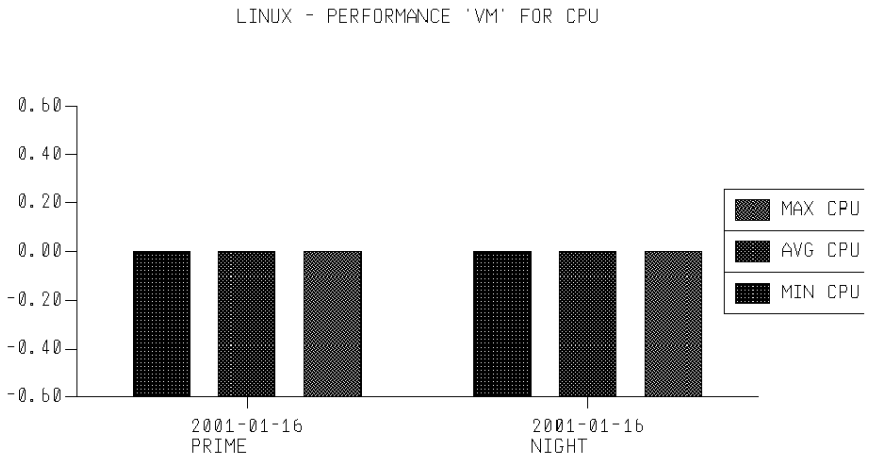


Figure 81. Example of a LINUX - Performance 'VM' for CPU report

The report contains the following information:

DATE Date of the measurement.

PERIOD
Name of the period.

MAX CPU
Maximum CPU value.

AVG CPU
Average CPU value.

MIN CPU
Minimum CPU value.

Reports in the Windows component

This section describes the Windows subcomponent reports:

- "Windows Disk Usage for System, Hourly Trend."
- "Windows Disk Usage for Device, Daily Overview" on page 193.
- "Windows CPU Utilization by System, Hourly Trend" on page 194.
- "Windows CPU Utilization by System, Daily Overview" on page 195.
- "Windows Memory Usage by System, Hourly Trend" on page 196.
- "Windows Memory Usage by System, Daily Overview" on page 197.
- "Windows System Overview Report" on page 198.

Windows Disk Usage for System, Hourly Trend

This report show the hourly trend of disk usage on a selected days for selected nodes. It allows you to identify potential periods during the day when unusually high disk usage is taking place.

This information identifies the report:

Report ID
WIN001

Report group
WINDOWS

Source
WIN_PERF_DD_H

Attributes
WINDOWS, DISK, DEVICE, TREND, FREE, SPACE

Variables
FROM_DATE, TO_DATE, NODE_NAME

Windows Disk Usage for System, Hourly Trend
Date: '2009-06-12' to '2009-06-12'

NODE NAME	DATE	DEVICE NAME	TIME	FREE SPACE MIN (MB)	FREE SPACE AVG (MB)	FREE SPACE MAX (MB)	PERCENT FREE MIN	PERCENT FREE AVG	PERCENT FREE MAX
IBM-PAXPRIM	2009-06-12	C:	00.00.00	61691	61691	61691	67	67	67
		C:	01.00.00	61684	61688	61691	67	67	67
		C:	02.00.00	61684	61684	61684	67	67	67
		C:	03.00.00	61684	61684	61684	67	67	67
		C:	04.00.00	61684	61684	61684	67	67	67
		C:	05.00.00	61684	61702	61720	67	67	67
		C:	06.00.00	61720	61720	61720	67	67	67
		C:	07.00.00	61720	61720	61720	67	67	67
		C:	08.00.00	61720	61720	61720	67	67	67
		C:	09.00.00	61720	61720	61720	67	67	67
		C:	10.00.00	61594	61595	61595	67	67	67
		C:	11.00.00	61587	61588	61588	67	67	67
		C:	12.00.00	61586	61587	61587	67	67	67
		C:	13.00.00	61584	61586	61587	67	67	67
		C:	14.00.00	61565	61585	61604	67	67	67
		C:	15.00.00	61634	61635	61635	67	67	67
		C:	16.00.00	61631	61633	61634	67	67	67
		C:	17.00.00	61629	61633	61636	67	67	67
		C:	18.00.00	61635	61635	61635	67	67	67
		C:	19.00.00	61633	61634	61634	67	67	67
		C:	20.00.00	61632	61632	61632	67	67	67
		C:	21.00.00	61631	61632	61632	67	67	67
		C:	22.00.00	61630	61631	61631	67	67	67
		C:	23.00.00	61629	61630	61630	67	67	67

Tivoli Decision Support: WIN001

Figure 82. Example of Windows Disk Usage for System, Hourly Trend

Windows Disk Usage for Device, Daily Overview

This report show the daily overview of disk usage for the selected nodes and devices, if any. It allows you to identify potential adverse changes in disk usage.

This information identifies the report:

Report ID
WIN002

Report group
WINDOWS

Source
WIN_PERF_DD_D

Attributes
WINDOWS, DISK, DEVICE, OVERVIEW, FREE, SPACE

Variables
NODE_NAME, DEVICE_NAME

Windows Disk Usage for Device, Daily Overview

DATE	NODE NAME	DEVICE NAME	FREE SPACE MIN (MB)	FREE SPACE AVG (MB)	FREE SPACE MAX (MB)	PERCENT FREE MIN	PERCENT FREE AVG	PERCENT FREE MAX
2009-06-12	IBM-PAXPRIM	C:	61565	61652	61720	67	67	67

Tivoli Decision Support: WIN002

Figure 83. Example of Windows Disk Usage for Device, Daily Overview

Windows CPU Utilization by System, Hourly Trend

This report show the hourly trend of CPU usage for the selected dates and nodes. It allows you to identify potential problems with processor usage.

This information identifies the report:

Report ID
WIN003

Report group
WINDOWS

Source
WIN_PERF_PU_H

Attributes
WINDOWS, CPU, UTILIZATION, TREND, FREE

Variables
FROM_DATE, TO_DATE, NODE_NAME

Windows CPU Utilization by System, Hourly Trend
Date: '2009-04-15' to '2009-06-12'

NODE NAME	DATE	TIME	CPU USAGE MIN PERCENT	CPU USAGE AVG PERCENT	CPU USAGE MAX PERCENT
IBM-PAXPRIM	2009-06-12	00.00.00	4	4	7
		01.00.00	4	5	5
		02.00.00	5	5	5
		03.00.00	5	5	5
		04.00.00	4	5	5
		05.00.00	4	4	5
		06.00.00	4	4	4
		07.00.00	4	4	4
		08.00.00	4	4	4
		09.00.00	4	4	5
		10.00.00	4	5	5
		11.00.00	5	5	5
		12.00.00	5	5	5
		13.00.00	5	5	5
		14.00.00	5	5	5
		15.00.00	5	5	5
		16.00.00	5	5	5
		17.00.00	5	5	6
		18.00.00	6	6	6
		19.00.00	6	7	7
		20.00.00	7	7	8
		21.00.00	8	8	8
		22.00.00	8	9	9
		23.00.00	9	9	9

Tivoli Decision Support: WIN003

Figure 84. Example of Windows CPU Utilization by System, Hourly Trend

Windows CPU Utilization by System, Daily Overview

This report show the daily overview of CPU usage for the selected dates and nodes. It allows you to identify potential problems with processor usage over the longer term.

This information identifies the report:

Report ID
WIN004

Report group
WINDOWS

Source
WIN_PERF_PU_D

Attributes
WINDOWS, CPU, UTILIZATION, OVERVIEW, FREE

Variables
NODE_NAME

Windows CPU Utilization by System, Daily Overview

NODE NAME	DATE	CPU USAGE MIN PERCENT	CPU USAGE AVG PERCENT	CPU USAGE MAX PERCENT
IBM-PAXPRIM	2009-06-12	4	5	9
IBM-PAXPRIM	2009-06-13	2	15	74
IBM-PAXPRIM	2009-06-14	1	16	79

Tivoli Decision Support: WIN004

Figure 85. Example of Windows CPU Utilization by System, Daily Overview

Windows Memory Usage by System, Hourly Trend

This report show the hourly trend of memory usage for the selected dates and nodes. It allows you to identify potential problems with memory usage during the day.

This information identifies the report:

Report ID
WIN005

Report group
WINDOWS

Source
WIN_PERF_PU_H

Attributes
WINDOWS, MEMORY, USAGE, TREND, FREE

Variables
FROM_DATE, TO_DATE, NODE_NAME

Windows Memory Usage by System, Hourly Trend
Date: '2009-06-12' to '2009-06-12'

NODE NAME	DATE	TIME	MEMORY FREE MIN (KB)	MEMORY FREE AVG (KB)	MEMORY FREE MAX (KB)
IBM-PAXPRIM	2009-06-12	00.00.00	1399924	1407371	1408564
		01.00.00	1364320	1380610	1407532
		02.00.00	1367096	1367813	1368684
		03.00.00	1366520	1367241	1367988
		04.00.00	1366172	1366796	1367520
		05.00.00	1346256	1380571	1389388
		06.00.00	1387428	1388094	1388876
		07.00.00	1376300	1387421	1388156
		08.00.00	1386156	1386948	1387664
		09.00.00	1015916	1353313	1399148
		10.00.00	1045336	1155013	1214188
		11.00.00	1015476	1056817	1120244
		12.00.00	1100632	1152908	1213140
		13.00.00	1062504	1138608	1200420
		14.00.00	920532	984165	1096668
		15.00.00	893760	988563	1047896
		16.00.00	924324	983061	1050484
		17.00.00	937800	958081	963300
		18.00.00	927856	941137	944116
		19.00.00	938352	940428	941956
		20.00.00	934356	939315	940936
		21.00.00	919056	937112	939584
		22.00.00	928324	933017	935268
		23.00.00	925836	932618	933872

Tivoli Decision Support: WIN005

Figure 86. Example of Windows Memory Usage by System, Hourly Trend

Windows Memory Usage by System, Daily Overview

This report show the daily overview of memory usage for the selected nodes. It allows you to identify potential problems with memory usage over the medium to long term.

This information identifies the report:

Report ID
WIN006

Report group
WINDOWS

Source
WIN_PERF_PU_D

Attributes
WINDOWS, MEMORY, USAGE, TREND, FREE

Variables
NODE_NAME

Windows Memory Usage by System, Daily Overview

NODE NAME	DATE	MEMORY FREE MIN (KB)	MEMORY FREE AVG (KB)	MEMORY FREE MAX (KB)
IBM-PAXPRIM	2009-06-12	893760	1159459	1408564

Tivoli Decision Support: WIN006

Figure 87. Example of Windows Memory Usage by System, Daily Overview

Windows System Overview Report

This report shows the system overview for a given data range and node list. It is intended to provide a high-level health check for all listed Windows nodes.

This information identifies the report:

Report ID
WIN007

Report group
WINDOWS

Source
WIN_PERF_PU_H, WIN_PERF_DD_H

Attributes
WINDOWS, SYSTEM, MEMORY, DISK, USAGE, TREND, OVERVIEW

Variables
FROM_DATE, TO_DATE, NODE_NAME

Windows System Overview Report
Date: '2009-04-15' to '2009-04-15'

NODE NAME	DATE	TIME	CPU USAGE MIN %	CPU USAGE AVG %	CPU USAGE MAX %	MEMORY FREE MIN (KB)	MEMORY FREE AVG (KB)	MEMORY FREE MAX (KB)	FREE SPACE MIN %	FREE SPACE AVG %	FREE SPACE MAX %	FREE SPACE MIN (MB)	FREE SPACE AVG (MB)	FREE SPACE MAX (MB)
IBM-PAXW2K3	2009-04-15	01.00.00	0	0	0	761436	767177	769700	6	6	6	71059	71059	71059
		02.00.00	0	0	0	762992	767326	768264	6	6	6	71059	71059	71059
		03.00.00	0	0	0	762488	767168	769344	6	6	6	71059	71059	71059
		04.00.00	0	0	0	763248	767604	769248	6	6	6	71059	71059	71059
		05.00.00	0	0	0	762536	767026	767936	6	6	6	71059	71059	71059
		06.00.00	0	0	0	750628	765030	768428	6	6	6	71059	71059	71059
		07.00.00	0	0	0	761988	766864	768684	6	6	6	71059	71059	71059
		08.00.00	0	0	0	761828	766428	767796	6	6	6	71059	71059	71059
		09.00.00	0	0	0	762660	766984	768732	6	6	6	71059	71059	71059
		10.00.00	0	0	0	761768	766314	767748	6	6	6	71059	71059	71059
		11.00.00	0	0	0	761848	767150	769156	6	6	6	71059	71059	71059
		12.00.00	0	0	0	763076	767396	769036	6	6	6	71059	71059	71059
		13.00.00	0	0	0	761764	766653	768648	6	6	6	71059	71059	71059
		14.00.00	0	0	0	762248	766901	768456	6	6	6	71059	71059	71059
		15.00.00	0	0	0	761992	766368	767444	6	6	6	71059	71059	71059
		16.00.00	0	0	0	759528	765585	767356	6	6	6	71059	71059	71059
		17.00.00	0	0	0	761256	765821	767588	6	6	6	71059	71059	71059
		18.00.00	0	0	0	760948	765652	767776	6	6	6	71059	71059	71059
		19.00.00	0	0	0	761504	765825	767204	6	6	6	71059	71059	71059
		20.00.00	0	0	0	760660	765440	767024	6	6	6	71059	71059	71059
		21.00.00	0	0	0	760556	765945	767820	6	6	6	71059	71059	71059
		22.00.00	0	0	0	761356	765693	767436	6	6	6	71059	71059	71059
		23.00.00	0	0	0	760828	766078	768404	6	6	6	71059	71059	71059

Tivoli Decision Support: WIN007

Figure 88. Example of Windows System Overview Report

Chapter 15. Log record definitions

This chapter describes (in alphabetical sequence) the following record definitions:

XACCT_COMMAND

This process accounting record definition is used for selecting records with record type PACCT.

Main section

Field name	Offset	Length	Format	Description
RECORD_LENGTH	0	2	BINARY	Record length
NODE_NAME	4	16	CHAR	Node name
RECORD_TYPE	21	5	CHAR	Record type
DTE	27	6	CHAR	Date
TME	33	6	TIME(HHMMSS)	Time
AC_FLAG	40	5	EXTERNAL INTEGER	Acct. flag
AC_STAT	46	5	EXTERNAL INTEGER	Exit status
AC_UID	52	8	CHAR	User name
AC_GID	61	8	CHAR	Group name
AC_TTY	70	10	CHAR	Control typewriter
AC_COMM	81	12	CHAR	Command name
AC_UTIME	94	10	EXTERNAL FLOAT	User time
AC_STIME	105	10	EXTERNAL FLOAT	System time
AC_ETIME	116	16	EXTERNAL FLOAT	Elapsed time
AC_MEM	133	10	EXTERNAL FLOAT	Memory usage
AC_IO	144	16	EXTERNAL FLOAT	Characters Transferred
AC_RW	161	10	EXTERNAL FLOAT	Blocks Read/Written

XACCT_CONNECT

This connect time accounting record definition is used for selecting records with record type WTMP.

Main section

Field name	Offset	Length	Format	Description
RECORD_LENGTH	0	2	BINARY	Record length
NODE_NAME	4	16	CHAR	Node name
RECORD_TYPE	21	5	CHAR	Record type
DTE	27	6	CHAR	Date
TME	33	6	TIME(HHMMSS)	Time
UT_USER	40	8	CHAR	User login name
UT_ID	49	14	CHAR	/etc/inittab ID
UT_LINE	64	12	CHAR	Device name
UT_TYPE	77	2	CHAR	Type of entry
UT_PID	80	6	EXTERNAL INTEGER	Connect ID
E_TERMINATION	87	2	CHAR	Connect term status
E_EXIT	90	2	CHAR	Connect exit status
UT_TIME	93	10	EXTERNAL FLOAT	Time entry was made (see Note)
UT_HOST	104	*	CHAR	Node name

Note:
Time is given in seconds, measured from 01.01.1970

XACCT_DISK

This disk accounting record definition is used for selecting records with record type DTMP.

Main section

Field name	Offset	Length	Format	Description
RECORD_LENGTH	0	2	BINARY	Record length
NODE_NAME	4	16	CHAR	Node name
RECORD_TYPE	21	5	CHAR	Record type
DTE	27	6	CHAR	Date
TME	33	6	TIME(HHMMSS)	Time
USERID	40	11	EXTERNAL INTEGER	User ID number
USER_NAME	52	8	CHAR	User name
DISK_BLOCKS	61	8	EXTERNAL INTEGER	Disk blocks

XACCT_PRINT

This print accounting record definition is used for selecting records with record type QACCT.

Main section

Field name	Offset	Length	Format	Description
RECORD_LENGTH	0	2	BINARY	Record length
NODE_NAME	4	16	CHAR	Node name
RECORD_TYPE	21	5	CHAR	Record type
DTE	27	6	CHAR	Date
TME	33	6	TIME(HHMMSS)	Time
PRINT_QUEUE	40	8	CHAR	Printer queue remote name
PAGES	49	8	EXTERNAL INTEGER	Printed pages
COPIES	58	8	EXTERNAL INTEGER	The number of printouts
PR_USER	67	*	CHAR(*)	User and requesting node

XCONFIG_HW

This hardware configuration record definition is used for selecting records with record type CONF and record subtypes A, D, or S.

Main section

Field name	Offset	Length	Format	Description
RECORD_LENGTH	0	2	BINARY	Record length
NODE_NAME	4	16	CHAR	Node name
RECORD_TYPE	21	5	CHAR	Record type
DTE	27	6	CHAR	Date
TME	33	6	TIME(HHMMSS)	Time
RECORD_SUBTYPE	40	2	CHAR	Record subtype
DEVICE_NAME	43	16	CHAR	Device name
LOCATION	60	16	CHAR	Location
DEVICE_CLASS	77	16	CHAR	Device class
DEVICE_TYPE	94	16	CHAR	Device type
DEVICE_SUBCLASS	111	16	CHAR	Device subclass
DEVICE_DESC	128	*	CHAR(*)	Device descr.

XCONFIG_SW

This software configuration record definition is used for selecting records with record type CONF and record subtypes H, I, or L.

Main section

Field name	Offset	Length	Format	Description
RECORD_LENGTH	0	2	BINARY	Record length
NODE_NAME	4	16	CHAR	Node name
RECORD_TYPE	21	5	CHAR	Record type
DTE	27	6	CHAR	Date
TME	33	6	TIME(HHMMSS)	Time
RECORD_SUBTYPE	40	2	CHAR	Record subtype

Record section SUBTYPE_H

About this task

Present if RECORD_SUBTYPE has the value H.
Offset 42
Length
36

Field name	Offset	Length	Format	Description
RELEASE	1	15	CHAR	Release number
STATUS	17	10	CHAR	Installation status
INST_DATE	28	8	CHAR	Installation date

Record section SUBTYPE_I

Present if RECORD_SUBTYPE has the value I.
Offset 42
Length
37

Field name	Offset	Length	Format	Description
PRODUCT_ID	1	10	CHAR	Product ID
FEATURE_ID	12	4	CHAR	Feature ID
PRODUCT_NAME	17	20	CHAR	Product name

Record section SUBTYPE_L

Present if RECORD_SUBTYPE has the value L.
Offset 42
Length
13

Field name	Offset	Length	Format	Description
STATE	1	12	CHAR	Product state

Record section FILE_NAME

Offset 79
Length
Length of FILE_NAME_L

Field name	Offset	Length	Format	Description
FILE_NAME_L	1	4	EXTERNAL INTEGER	Length of section FILE_NAME
FILE_NAME	6	*	CHAR(*)	File name

Record section PATH_NAME

Offset Dependent upon length of previous section
Length
Length of PATH_NAME_L

Field name	Offset	Length	Format	Description
PATH_NAME_L	1	4	EXTERNAL INTEGER	Length of section PATH_NAME
PATH_NAME	6	*	CHAR(*)	Path name

Record section PRODUCT_DESC

Present if RECORD_SUBTYPE has the value L.
Offset Dependent upon length of previous section
Length
Length of PRODUCT_DESC_L

Field name	Offset	Length	Format	Description
PRODUCT_DESC_L	1	4	EXTERNAL INTEGER	Length of section PRODUCT_DESC
PRODUCT_DESC	6	*	CHAR(*)	Product description

XERROR

This error record definition is used for selecting records with record type ERRPT.

Main section

Field name	Offset	Length	Format	Description
RECORD_LENGTH	0	2	BINARY	Record length
NODE_NAME	4	16	CHAR	Node name
RECORD_TYPE	21	5	CHAR	Record type
DTE	27	6	CHAR	Date
TME	33	6	TIME(HHMMSS)	Time
ERROR_ID	40	8	CHAR	Error ID
ERROR_TYPE	49	1	CHAR	Error type
ERROR_CLASS	51	1	CHAR	Error class
RESOURCE_NAME	53	14	CHAR	Resource name
DESCRIPTION	68	*	CHAR(*)	Error descr.

XPERF_CPU

This CPU performance record definition is used for selecting records with record type PERF and record subtype VM.

Main section

Field name	Offset	Length	Format	Description
RECORD_LENGTH	0	2	BINARY	Record length
NODE_NAME	4	16	CHAR	Node name
RECORD_TYPE	21	5	CHAR	Record type
DTE	27	6	CHAR	Date
TME	33	6	TIME(HHMMSS)	Time
RECORD_SUBTYPE	40	2	CHAR	Record subtype
INTERVAL	43	4	EXTERNAL INTEGER	Measured time
MEMORY_MIN	48	9	EXTERNAL INTEGER	Free pages, min
MEMORY_AVG	58	11	EXTERNAL FLOAT	Free pages, avg
MEMORY_MAX	70	9	EXTERNAL INTEGER	Free pages, max
PAGING_MIN	80	7	EXTERNAL INTEGER	Paging rate, min
PAGING_AVG	88	9	EXTERNAL FLOAT	Paging rate, avg
PAGING_MAX	98	7	EXTERNAL INTEGER	Paging rate, max
CPU_MIN	106	7	EXTERNAL INTEGER	CPU usage, min
CPU_AVG	114	9	EXTERNAL FLOAT	CPU usage, avg
CPU_MAX	124	7	EXTERNAL INTEGER	CPU usage, max
USERS	132	5	EXTERNAL INTEGER	Number of users
PROCESSES	138	5	EXTERNAL INTEGER	Number of processes

XPERF_DISK_FS

This disk space performance record definition is used for selecting records with record type DISK and record subtype FS.

Main section

Field name	Offset	Length	Format	Description
RECORD_LENGTH	0	2	BINARY	Record length
NODE_NAME	4	16	CHAR	Node name
RECORD_TYPE	21	5	CHAR	Record type
DTE	27	6	CHAR	Date
TME	33	6	TIME(HHMMSS)	Time
RECORD_SUBTYPE	40	2	CHAR	Record subtype

Record section FILE_SYSTEM

Offset 42

Length
24

Field name	Offset	Length	Format	Description
TOTAL_SPACE_KB	1	9	EXTERNAL INTEGER	Total space, in kilobytes
FREE_SPACE_KB	11	9	EXTERNAL INTEGER	Free space, in kilobytes
USED_SPACE_PCT	21	3	EXTERNAL INTEGER	Used space, in percent

Record section DEVICE_NAME

Offset 66
Length
Length of DEVICE_NAME_L

Field name	Offset	Length	Format	Description
DEVICE_NAME_L	1	4	EXTERNAL INTEGER	Length of section DEVICE_NAME
DEVICE_NAME	6	*	CHAR(*)	Device name

Record section SPACE_NAME

Offset 66
Length
Length of SPACE_NAME_L

Field name	Offset	Length	Format	Description
SPACE_NAME_L	1	4	EXTERNAL INTEGER	Length of section SPACE_NAME
SPACE_NAME	6	*	CHAR(*)	Space name

XPERF_DISK_VG

This disk space performance record definition is used for selecting records with record type DISK and record subtype VG.

Main section

Field name	Offset	Length	Format	Description
RECORD_LENGTH	0	2	BINARY	Record length
NODE_NAME	4	16	CHAR	Node name
RECORD_TYPE	21	5	CHAR	Record type
DTE	27	6	CHAR	Date
TME	33	6	TIME(HHMMSS)	Time
RECORD_SUBTYPE	40	2	CHAR	Record subtype

Record section VOLUME_GROUP

Offset 42
Length
30

Field name	Offset	Length	Format	Description
TOTAL_SPACE_MB	1	9	EXTERNAL INTEGER	Total space, in megabytes
FREE_SPACE_MB	11	9	EXTERNAL INTEGER	Free space, in megabytes
USED_SPACE_MB	21	9	EXTERNAL INTEGER	Used space, in megabytes

Record section SPACE_NAME

Offset Dependent upon length of previous section
Length
Length of SPACE_NAME_L

Field name	Offset	Length	Format	Description
SPACE_NAME_L	1	4	EXTERNAL INTEGER	Length of section SPACE_NAME
SPACE_NAME	6	*	CHAR(*)	Space name

XPERF_DISKIO

This disk I/O performance record definition is used for selecting records with record type PERF and record subtype IO.

Main section

Field name	Offset	Length	Format	Description
RECORD_LENGTH	0	2	BINARY	Record length
NODE_NAME	4	16	CHAR	Node name
RECORD_TYPE	21	5	CHAR	Record type
DTE	27	6	CHAR	Date
TME	33	6	TIME(HHMMSS)	Time
RECORD_SUBTYPE	40	2	CHAR	Record subtype
INTERVAL	43	4	EXTERNAL INTEGER	Measurement interval
MINREAD	48	9	EXTERNAL INTEGER	KB read, minimum
AVGREAD	58	9	EXTERNAL FLOAT	KB read, average
MAXREAD	68	9	EXTERNAL INTEGER	KB read, maximum
TOTREAD	78	8	EXTERNAL INTEGER	KB read, total
MINWRITE	87	7	EXTERNAL INTEGER	KB write, minimum
AVGWRITE	95	9	EXTERNAL FLOAT	KB write, average
MAXWRITE	105	7	EXTERNAL INTEGER	KB write, maximum
TOTWRITE	113	8	EXTERNAL INTEGER	KB write, total
MINRW	122	7	EXTERNAL INTEGER	KB read and written, minimum
AVGRW	130	9	EXTERNAL FLOAT	KB read and written, average

Field name	Offset	Length	Format	Description
MAXRW	140	7	EXTERNAL INTEGER	KB read and written, maximum
TOTRW	148	8	EXTERNAL INTEGER	KB read and written, total
BUSY	157	5	EXTERNAL FLOAT	Busy %
DISK	163	*	CHAR(*)	Physical disk name

XPERF_PAGING

This paging space performance record definition is used for selecting records with record type PERF and record subtype PS.

Main section

Field name	Offset	Length	Format	Description
RECORD_LENGTH	0	2	BINARY	Record length
NODE_NAME	4	16	CHAR	Node name
RECORD_TYPE	21	5	CHAR	Record type
DTE	27	6	CHAR	Date
TME	33	6	TIME(HHMMSS)	Time
RECORD_SUBTYPE	40	2	CHAR	Record subtype
PAGE_SPACE	43	20	CHAR	Page space name
DISK	64	20	CHAR	Physical disk name
PSSIZE	85	5	EXTERNAL INTEGER	Page space size in KB
PSUSE	91	5	EXTERNAL INTEGER	Page space used in KB

WIN_PERF_DD

This record gives the device information for each non-removable device on the machine. The information is gathered with the following WMI command:

```
select Name, Size, FreeSpace
    from Win32_LogicalDisk
    where DriveType = 3
```

This is both gathered and reported based on the configuration item xa_XPerfDiskFsRun. Note that there is a single space following each field.

Main section

Column	Position	Size	Description
Type	0	2	DD to indicate this is a disk device record.
Date	3	8	YYYYMMDD of the record.
Time	12	6	HHMMSS of the record.
Node name	19	16	Node name of the machine.
Device name	36	8	Device name.
Total size	45	9	Zero-padded device size in megabytes (10242 bytes).

Column	Position	Size	Description
Free space	55	9	Zero-padded device free space in megabytes.
Free %	65	3	Zero-padded percentage free.

WIN_PERF_PU

This record gives the minimum, maximum and average figures for CPU percentage use and memory usage. The information is gathered with the following WMI commands:

```
select PercentProcessorTime, TimeStamp_100NS
    from Win32_PerfRawData_PerfOS_Processor
    where Name = '_Total'
select TotalPhysicalMemory
    from Win32_LogicalMemoryConfiguration
select AvailableKBytes
    from Win32_PerfFormattedData_PerfOS_Memory
```

This is gathered continuously and reported (aggregated) based on the configuration item xb_XPerfCpuRun. Note that there is a single space following each field.

Main section

Column	Posit ion	Size	Description
Type	0	2	PU to indicate this is a performance usage record.
Date	3	8	YYYYMMDD of the record.
Time	12	6	HHMMSS of the record.
Node name	19	16	Node name of the machine.
CPU count	36	2	Number of CPUs in this system.
Average CPU usage	39	3	Zero-padded average CPU percentage usage for period. This is the average of all the snapshots in the period.
Minimum CPU usage	43	3	Zero-padded minimum CPU percentage usage for period. A snapshot CPU usage figure is calculated by sampling all of the CPUs and averaging their usage (so a 2-CPU system running at 70% and 30% gives 50%). These snapshots are collected as often as possible but only reported based on the xb_XPerfCpuRun configuration item. This is the minimum snapshot for the period.
Maximum CPU usage	47	3	Zero-padded maximum CPU percentage usage for period. This is the minimum snapshot for the period.
Memory size	51	9	Zero-padded total memory size at end of period, in kilobytes.
Average memory	61	9	Zero-padded average memory free for period, in kilobytes.
Minimum memory	71	9	Zero-padded minimum memory free for period, in kilobytes (1024 bytes).
Maximum memory	81	9	Zero-padded maximum memory free for period, in kilobytes.

Part 5. Appendixes

Appendix. Support information

If you have a problem with your IBM software, you want to resolve it quickly. This section describes the following options for obtaining support for IBM software products:

- "Searching knowledge bases"
- "Obtaining fixes"
- "Receiving weekly support updates" on page 214
- "Contacting IBM Software Support" on page 214

Searching knowledge bases

You can search the available knowledge bases to determine whether your problem was already encountered and is already documented.

Searching the information center

IBM provides extensive documentation that can be installed on your local computer or on an intranet server. You can use the search function of this information center to query conceptual information, instructions for completing tasks, and reference information.

Searching the Internet

If you cannot find an answer to your question in the information center, search the Internet for the latest, most complete information that might help you resolve your problem.

To search multiple Internet resources for your product, use the **Web search** topic in your information center. In the navigation frame, click **Troubleshooting and support** ▸ **Searching knowledge bases** and select **Web search**. From this topic, you can search a variety of resources, including the following:

- IBM technotes
- IBM downloads
- IBM developerWorks®
- Forums and newsgroups
- Google

Obtaining fixes

A product fix might be available to resolve your problem. To determine what fixes are available for your IBM software product, follow these steps:

1. Go to the IBM Software Support Web site at http://www.ibm.com/software/support/.
2. Click **Downloads and drivers** in the **Support topics** section.
3. Select the **Software** category.
4. Select a product in the **Sub-category** list.
5. In the **Find downloads and drivers by product** section, select one software category from the **Category** list.
6. Select one product from the **Sub-category** list.

7. Type more search terms in the **Search within results** if you want to refine your search.
8. Click **Search**.
9. From the list of downloads returned by your search, click the name of a fix to read the description of the fix and to optionally download the fix.

For more information about the types of fixes that are available, see the *IBM Software Support Handbook* at http://www-304.ibm.com/support/customercare/sas/f/handbook/home.html.

Receiving weekly support updates

To receive weekly e-mail notifications about fixes and other software support news, follow these steps:

1. Go to the IBM Software Support Web site at http://www.ibm.com/support/us/.
2. Click **My support** in the upper right corner of the page.
3. If you have already registered for **My support**, sign in and skip to the next step. If you have not registered, click **register now**. Complete the registration form using your e-mail address as your IBM ID and click **Submit**.
4. Click **Edit profile**.
5. In the **Products** list, select **Software**. A second list is displayed.
6. In the second list, select a product segment, for example, **Application servers**. A third list is displayed.
7. In the third list, select a product sub-segment, for example, **Distributed Application & Web Servers**. A list of applicable products is displayed.
8. Select the products for which you want to receive updates, for example, **IBM HTTP Server** and **WebSphere® Application Server**.
9. Click **Add products**.
10. After selecting all products that are of interest to you, click **Subscribe to email** on the **Edit profile** tab.
11. Select **Please send these documents by weekly email**.
12. Update your e-mail address as needed.
13. In the **Documents** list, select **Software**.
14. Select the types of documents that you want to receive information about.
15. Click **Update**.

If you experience problems with the **My support** feature, you can obtain help in one of the following ways:

Online
Send an e-mail message to erchelp@ca.ibm.com, describing your problem.

By phone
Call 1-800-IBM-4You (1-800-426-4968).

Contacting IBM Software Support

IBM Software Support provides assistance with product defects.

Before contacting IBM Software Support, your company must have an active IBM software maintenance contract, and you must be authorized to submit problems to IBM. The type of software maintenance contract that you need depends on the type of product you have:

- For IBM distributed software products (including, but not limited to, Tivoli, Lotus®, and Rational® products, as well as DB2 and WebSphere products that run on Windows, or UNIX operating systems), enroll in Passport Advantage® in one of the following ways:

  Online

  Go to the Passport Advantage Web site at http://www.lotus.com/services/passport.nsf/ WebDocs/Passport_Advantage_Home and click **How to Enroll**.

  By phone

  For the phone number to call in your country, go to the IBM Software Support Web site at http://techsupport.services.ibm.com/guides/contacts.html and click the name of your geographic region.
- For customers with Subscription and Support (S & S) contracts, go to the Software Service Request Web site at https://techsupport.services.ibm.com/ssr/login.
- For customers with IBMLink, CATIA, Linux, S/390®, iSeries, pSeries, zSeries, and other support agreements, go to the IBM Support Line Web site at http://www.ibm.com/services/us/index.wss/so/its/a1000030/dt006.
- For IBM eServer™ software products (including, but not limited to, DB2 and WebSphere products that run in zSeries, pSeries, and iSeries environments), you can purchase a software maintenance agreement by working directly with an IBM sales representative or an IBM Business Partner. For more information about support for eServer software products, go to the IBM Technical Support Advantage Web site at http://www.ibm.com/servers/eserver/techsupport.html.

If you are not sure what type of software maintenance contract you need, call 1-800-IBMSERV (1-800-426-7378) in the United States. From other countries, go to the contacts page of the *IBM Software Support Handbook* on the Web at http://techsupport.services.ibm.com/guides/contacts.html and click the name of your geographic region for phone numbers of people who provide support for your location.

To contact IBM Software support, follow these steps:

1. "Determining the business impact"
2. "Describing problems and gathering information" on page 216
3. "Submitting problems" on page 216

Determining the business impact

When you report a problem to IBM, you are asked to supply a severity level. Therefore, you need to understand and assess the business impact of the problem that you are reporting. Use the following criteria:

Severity 1

The problem has a *critical* business impact. You are unable to use the program, resulting in a critical impact on operations. This condition requires an immediate solution.

Severity 2
The problem has a *significant* business impact. The program is usable, but it is severely limited.

Severity 3
The problem has *some* business impact. The program is usable, but less significant features (not critical to operations) are unavailable.

Severity 4
The problem has *minimal* business impact. The problem causes little impact on operations, or a reasonable circumvention to the problem was implemented.

Describing problems and gathering information

When describing a problem to IBM, be as specific as possible. Include all relevant background information so that IBM Software Support specialists can help you solve the problem efficiently. To save time, know the answers to these questions:

- What software versions were you running when the problem occurred?
- Do you have logs, traces, and messages that are related to the problem symptoms? IBM Software Support is likely to ask for this information.
- Can you re-create the problem? If so, what steps were performed to re-create the problem?
- Did you make any changes to the system? For example, did you make changes to the hardware, operating system, networking software, and so on.
- Are you currently using a workaround for the problem? If so, be prepared to explain the workaround when you report the problem.

Submitting problems

You can submit your problem to IBM Software Support in one of two ways:

Online
Click **Submit and track problems** on the IBM Software Support site at http://www.ibm.com/software/support/probsub.html. Type your information into the appropriate problem submission form.

By phone
For the phone number to call in your country, go to the contacts page of the *IBM Software Support Handbook* at http://techsupport.services.ibm.com/guides/contacts.html and click the name of your geographic region.

If the problem you submit is for a software defect or for missing or inaccurate documentation, IBM Software Support creates an Authorized Program Analysis Report (APAR). The APAR describes the problem in detail. Whenever possible, IBM Software Support provides a workaround that you can implement until the APAR is resolved and a fix is delivered. IBM publishes resolved APARs on the Software Support Web site daily, so that other users who experience the same problem can benefit from the same resolution.

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement might not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to

IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

If you are viewing this information in softcopy form, the photographs and color illustrations might not display.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Bibliography

TDS publications

Administration Guide and Reference, SH19-6816, SH19-6816

Resource Accounting for z/OS, SH19-4495, SH19-4495

AS/400 System Performance Feature Guide and Reference, SH19-4019, SH19-4019

CICS Performance Feature Guide and Reference, SH19-6820, SH19-6820

Distributed Systems Performance Feature Guide and Reference, SH19-4018, SH19-4018

Guide to Reporting, SH19-6842, SH19-6842

IMS Performance Feature Guide and Reference, SH19-6825, SH19-6825

Language Guide and Reference, SH19-6817, SH19-6817

Messages and Problem Determination, SH19-6902, SH19-6902

Network Performance Feature Installation and Administration, SH19-6901, SH19-6901

Network Performance Feature Reference, SH19-6822, SH19-6822

Network Performance Feature Reports, SH19-6821, SH19-6821

Resource Accounting, SH19-6818, SH19-6818

System Performance Feature Guide, SH19-6819, SH19-6819

System Performance Feature Reference Volume I, SH19-4494, SH19-4494

System Performance Feature Reference Volume II, SC23-7966, SC23-7966

Glossary

administration
A Tivoli Decision Support for z/OS task that includes maintaining the database, updating environment information, and ensuring the accuracy of data collected.

administration dialog
The set of host windows used to administer Tivoli Decision Support for z/OS.

agent In UNIX Performance featureUNIX Performance feature, a set of programs running on a network node that gather data about the node and write the data to log files. Agents can gather data by issuing UNIX commands, and by reading files such as UNIX accounting files.

COLLECT
A process used by Tivoli Decision Support for z/OS to read data from input log data sets, interpret records in the data set, and store the data in DB2 tables in the Tivoli Decision Support for z/OS database.

component
An optionally installable part of an Tivoli Decision Support for z/OS feature. Specifically in Tivoli Decision Support for z/OS, a component refers to a logical group of objects used to collect log data from a specific source, to update the Tivoli Decision Support for z/OS database using that data, and to create reports from data in the database.

control table
A predefined Tivoli Decision Support for z/OS table that controls results returned by some log collector functions.

Data Language/I (DL/I)
An IBM database-management facility.

data table
A Tivoli Decision Support for z/OS table that contains performance data used to create reports.

distributed transaction processing
The distribution of processing among transactions that communicate synchronously with each other over intersystem or interregion links.

environment information
All of the information that is added to the log data to create reports. This information can include data such as performance groups, shift periods, installation definitions, and so on.

environment data
See “environment information.”

gather A process performed by agents in which performance data is entered into log files. The log files are located on the same nodes as the agents.

help topics
An online table of contents for the Tivoli Decision Support for z/OS online help information.

key columns
The columns of a DB2 table that together constitute the key.

key value
Value used to sort records into groups.

log Any sequential data set used as input to Tivoli Decision Support for z/OS.

log collector
A Tivoli Decision Support for z/OS program that processes log data sets and provides other services.

log collector language
Tivoli Decision Support for z/OS statements used to supply definitions to and invoke services of the log collector.

log data set
Any sequential data set used as input to Tivoli Decision Support for z/OS.

log definition
The description of a log data set processed by the log collector.

log procedure
A program module that is used to process all record types in certain log data sets.

logical unit (LU)
A port through which a user gains access to the services of a network.

lookup expression
An expression that specifies how a value is obtained from a lookup table.

lookup table
A Tivoli Decision Support for z/OS DB2 table that contains grouping, translation, or substitution information.

object An integral part of a feature component needed for data collection (for example, record definitions, record procedures, and update definitions).

process accounting
An analysis of how each process uses the processing unit, memory, and I/O resources

Tivoli Decision Support for z/OS database
A set of DB2 tables that includes data tables, lookup tables, system tables, and control tables.

purge condition
Instruction for purging unneeded data from the Tivoli Decision Support for z/OS database.

record definition
The description of a record type contained in the log data sets used by Tivoli Decision Support for z/OS, including detailed record layout and data formats.

record procedure
A program module that is called to process some types of log records.

record type
The classification of records in a log data set.

region A section of the dynamic area that is allocated to a job step or system task.

report definition language
Tivoli Decision Support for z/OS statements used to define reports and report groups.

report group
A collection of Tivoli Decision Support for z/OS reports that can be referred to by a single name.

reporting dialog
A set of host or workstation windows used to request reports.

resource
Any facility of the computing system or operating system required by a job or task, including central storage, input/output devices, the processing unit, data sets, and control or processing programs.

resource group
A collection of resources identified as belonging to a particular department or division. Resources are organized into groups to reflect the structure of an organization.

resource information
Environment information that describes the elements in a system (for example, a network).

section
A structure within a record that contains one or more fields and may contain other sections.

source In an update definition, the record or DB2 table that contains the data used to update an Tivoli Decision Support for z/OS DB2 table.

subcomponent
An optionally installable part of an Tivoli Decision Support for z/OS feature component.

system table
A DB2 table that stores information for controlling log collector processing, Tivoli Decision Support for z/OS dialogs, and reporting.

target In an update definition, the DB2 table in which Tivoli Decision Support for z/OS stores data from the source record or table.

threshold
The maximum or minimum acceptable level of usage. Usage measurements are compared with threshold levels.

Transmission Control Protocol/Internet Protocol (TCP/IP)
A non-proprietary communications protocol for linking workstations to host computers and to other hardware.

update definition
Instructions for entering data into DB2 tables from records of different types or from other DB2 tables.

updates
Instructions in Tivoli Decision Support for z/OS on how to process data from log data sets to DB2 tables.

view An alternative representation of data from one or more tables. A view can include all or some of the columns contained in the table on which it is defined.

Index

E

F

G

H

I

J

K

S

T

U

V

W

X

Z

Printed in USA

SH19-4018-13