IBM Tivoli Decision Support for z/OS
Version 1.8.2

Distributed Systems Performance
Feature Guide and Reference
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.

© COPYRIGHT INTERNATIONAL BUSINESS MACHINES CORPORATION. ALL RIGHTS RESERVED. 1994, 2015
Contents

Figures . . . . . . . . . . . . . . . . . . . . . . . . ix

Preface . . . . . . . . . . . . . . . . . . . . . . . xi
Who should read this book . . . . . . . . . . . . xi
What this book contains . . . . . . . . . . . . xi
Publications . . . . . . . . . . . . . . . . . . . . xiii
Tivoli Decision Support for z/OS library . . xiii
Accessing terminology online . . . . . . . . xiv
Accessing publications online . . . . . . . . xiv
Accessibility . . . . . . . . . . . . . . . . . . . . xv
Tivoli technical training . . . . . . . . . . . . xv
Support information . . . . . . . . . . . . . . . xv
Conventions used in this book . . . . . . . . . xv
Typeface conventions . . . . . . . . . . . . . . xv

Part 1. Distributed Systems
Performance Feature Guide . . . . . . . 1

Chapter 1. Introducing the Distributed
Systems Performance feature . . . . . . . 3
Gathering and collecting performance data . . 4
Report Groups . . . . . . . . . . . . . . . . . . 5

Chapter 2. Installing and configuring the
Distributed component . . . . . . . . . . . . 7
Planning the implementation process . . . . . 7
Installing and configuring your distributed nodes . 7
Installing and configuring your SLES nodes . . 8
Step 1: Checking requirements (SLES) . . . . . 8
Disk space requirements . . . . . . . . . . . . 8
Software requirements . . . . . . . . . . . . . 9
Step 2: Preparing and installing Tivoli Decision
Support for z/OS code on the distributed node
(SLES) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9
Step 3: Configuring the distributed node (SLES) . 10
Step 4: Starting the distributed node (SLES) . . 11
Step 5: Configuring auto-start on the distributed
node (SLES) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12
Installing and configuring your RHEL nodes . . 12
Step 1: Checking requirements (RHEL) . . . . . 12
Disk space requirements . . . . . . . . . . . . 12
Software requirements . . . . . . . . . . . . . 13
Step 2: Preparing and installing Tivoli Decision
Support for z/OS code on the distributed node
(RHEL) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13
Step 3: Configuring the distributed node (RHEL) . 14
Step 4: Starting the distributed node (RHEL) . . 15
Step 5: Configuring auto-start on the distributed
node (RHEL) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 16
Installing and configuring your HP-UX nodes . . 16
Step 1: Checking requirements (HP-UX) . . . . . 16
Disk space requirements . . . . . . . . . . . . 17
Software requirements . . . . . . . . . . . . . 17

Step 2: Preparing and installing Tivoli Decision
Support for z/OS code on the distributed node
(HP-UX) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 17
Step 3: Configuring the distributed node
(HP-UX) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19
Step 4: Starting the distributed node (HP-UX) . . 20
Step 5: Configuring auto-start on the distributed
node (HP-UX) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 20
Installing and configuring your AIX nodes . . . 21
Step 1: Checking requirements (AIX) . . . . . . 21
Disk space requirements . . . . . . . . . . . . 21
Software requirements . . . . . . . . . . . . . 21
Step 2: Preparing and installing Tivoli Decision
Support for z/OS code on the distributed node
(AIX) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 22
Step 3: Configuring the distributed node (AIX) . . 23
Step 4: Starting the distributed node (AIX) . . . 24
Step 5: Configuring auto-start on the distributed
node (AIX) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 24
Installing and configuring your Solaris nodes . . 25
Step 1: Check requirements (Solaris) . . . . . . 25
Disk space requirements . . . . . . . . . . . . 25
Software requirements . . . . . . . . . . . . . 26
Step 2: Preparing and installing Tivoli Decision
Support for z/OS code on the distributed node
(Solaris) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 26
Step 3: Configure the distributed node (Solaris) . 27
Step 4: Start and check the distributed node
(Solaris) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 28
Step 5: Configure auto-start on the distributed
node (Solaris) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 29
Installing Distributed component on the z/OS
system . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 29
Testing the installation . . . . . . . . . . . . . . 31
Putting the component into production . . . . . 32

Chapter 3. Installing and configuring
the Windows component . . . . . . . . . . . 33
Planning the implementation process . . . . . 33
Considering which Windows subcomponents to
install . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 34
Installing the Windows component on your
Windows nodes . . . . . . . . . . . . . . . . . . . . . . . . . . 35
Step 1: Checking Windows requirements . . . . . 35
Disk space requirements . . . . . . . . . . . . 35
Software requirements . . . . . . . . . . . . . 35
Step 2: Transferring Tivoli Decision Support for
z/OS feature code to Windows . . . . . . . . . . . . . . . . . 35
Capturing Windows component data and
transmitting to z/OS . . . . . . . . . . . . . . . . . . . . . . . . 36
Step 1: Customizing the Windows node . . . . . . 37
Parameter file values to change . . . . . . . . . . 37
Step 2: Setting up scheduled tasks . . . . . . . . . 37
Installing Windows component on the z/OS system 38
Collecting data under Tivoli Decision Support for z/OS .......................... 40
General collect job .................................................. 40
Testing the installation .............................................. 41
Putting the component into production ............................ 41

Part 2. Distributed Systems Performance Feature Reference .......................... 43

Chapter 4. Data flow and Tivoli Decision Support for z/OS objects .................. 45
Distributed Systems Performance feature general data flow ........................................ 45

Chapter 5. Data tables ........................................................................... 47
Naming standard for tables ......................................................... 47
Table descriptions ........................................................................ 47
Tables in the Distributed component .............................................. 48
DIST_CPU_H_.._M ........................................................... 48
DIST_DISK_D_.._M ......................................................... 49
DIST_DISKIO_H_.._D ...................................................... 49
DIST_PAGING_H_.._D_.._M ............................................ 50
Tables in the Windows component .................................................. 51
WIN_PERF_DD_H_.._D_.._M ........................................... 51
WIN_PERF_PU_H_.._D_.._M ........................................... 51

Chapter 6. Reports .................................................................................. 53
Report format and general description .................................................... 53
Report ID .............................................................................. 53
Report group ............................................................................ 53
Source .................................................................................... 54
Attributes .................................................................................. 54
Variables ................................................................................... 54
Reports in the Distributed component .................................................. 54
Distrib. CPU Util. for System, Hourly Trend ........................................... 54
Distrib. CPU Util. by System, Daily Overview ......................................... 55
Distrib. Statistics for System, Hourly Trend ........................................... 56
Distrib. Statistics by System, Daily Overview ......................................... 58
Distrib. File Systems, Daily Overview .................................................. 59
Distrib. Disk I/O for Disk, Hourly Trend ............................................... 60
Distrib. Disk I/O for System, Daily Overview .......................................... 61
Distrib. Page Space Util., Hourly Trend ............................................... 62
Distrib. CPU Util. Exceptions ......................................................... 63
Reports in the Windows component ..................................................... 64
Windows Disk Usage for System, Hourly Trend ........................................ 65
Windows Disk Usage for Device, Daily Overview ....................................... 65
Windows CPU Utilization by System, Hourly Trend .................................... 66
Windows CPU Utilization by System, Daily Overview ................................. 67
Windows Memory Usage by System, Hourly Trend .................................... 68
Windows Memory Usage by System, Daily Overview .................................. 69
Windows System Overview Report ..................................................... 70

Chapter 7. Log record definitions ............................................................ 73
DIST_PERF_VM_A ............................................................ 73
Main section .............................................................................. 73

DIST_DISK_FS_A ............................................................ 74
Main section .............................................................................. 74
Record section DEVICE ...................................................... 74
Record section SPACE ...................................................... 74
DIST_PERF_IO_A ............................................................ 74
Main section .............................................................................. 74
DIST_PERF_PS_A ............................................................ 75
Main section .............................................................................. 75
WIN_PERF_DD ................................................................. 76
Main section .............................................................................. 76
WIN_PERF_PU ................................................................. 76
Main section .............................................................................. 76

Part 3. Distributed Systems Performance Feature Guide (heritage) ................. 79

Chapter 8. Introducing the Distributed Systems Performance feature .................. 81
Gathering and collecting performance data ............................................... 81
Report groups ............................................................................. 82

Chapter 9. Installing and configuring the UNIX and Linux Performance components ........................................ 85
Planning the implementation process ..................................................... 85
Considering which UNIX subcomponents to install .................................... 86
Installing the UNIX Performance component on your UNIX nodes ............ 87
Step 1: Check UNIX requirements ..................................................... 87
Disk space requirements ............................................................... 87
Software requirements ................................................................. 87
Step 2: Transfer Tivoli Decision Support for z/OS feature code to UNIX ......... 88
Capturing UNIX Performance component data and transmitting to OS/390 .... 90
Step 1: Set up UNIX accounting ....................................................... 91
Setting up accounting for AIX ....................................................... 91
Setting up accounting for HP ....................................................... 93
Setting up accounting for Sun Solaris ................................................ 94
Step 2: Create log files ................................................................... 94
Step 3: Define the receiver of UNIX Performance component data ............ 95
Step 4: Customizing UNIX Performance component parameters ................ 96
Parameter file values for all subcomponents ......................................... 96
Parameter file values for Performance component .................................. 97
Step 5: Start the drlperfd daemon ..................................................... 98
Step 6: Invoke drlperfd at start time ................................................ 99
Description of daily tasks performed by UNIX Performance component .... 99
Operating the UNIX Performance component on UNIX ......................... 100
Description of UNIX Performance component files ............................... 101
AIX commands to use if UNIX Performance component installed ........... 102
drlperfd ................................................................. 102
Installing UNIX Performance component on the 
z/OS system ........................................ 102
Updating the lookup tables ........................................ 104
  Updating XACCT_CONNECT_TYPE ....... 105
  Updating X_NODE_NAME ...................... 105
  Updating XACCT_PRINT_TYPE .......... 105
Collecting data under Tivoli Decision Support for 
z/OS ........................................ 105
Creating your Generation Data Groups ........................................ 106
Renaming file sets sent from UNIX for use with a GDG ........................................ 106
  Example of renaming UNIX files ........................................ 106
General collect job ........................................ 107
Collect job with specified Tivoli Decision Support ........................................ 108
Support for z/OS feature code ........................................ 108
Backup and delete of generation data sets ........................................ 109
Testing the installation ........................................ 109
Putting the feature into production ........................................ 109
Considering which Linux subcomponents to install ........................................ 110
Installing the Linux Performance component on your Linux nodes ........................................ 110
  Step 1: Check Linux requirements ........ 110
  Software requirements .................. 110
  Step 2: Transfer Tivoli Decision Support for 
z/OS feature code to Linux ............... 110
Capturing Linux Performance component data and 
transmitting to OS/390 ........................................ 111
  Step 1: Create log files ................. 112
  Step 2: Define the receiver of Linux Performance component data .............. 112
  Step 3: Customizing Linux Performance component parameters ........ 113
    Parameter file values for all subcomponents ........................................ 113
    Parameter file values for performance component ........................................ 114
  Step 4: Start the drlperfd daemon ........................................ 115
Commands to use if Linux Performance component 
installed ........................................ 115
drlactperfd ........................................ 115
drldeactperfd ........................................ 115
drlsend ........................................ 116
Collecting data under Tivoli Decision Support for 
z/OS ........................................ 116
  General collect job ..................... 116
Part 4. Distributed Systems
Performance Feature Reference (heritage) ................. 127
Chapter 11. Data flow and Tivoli Decision Support for z/OS objects ........ 129
  Distributed Systems Performance component general data flow .............. 129
  Description of log files, record and log definitions ........................................ 131
Chapter 12. Data, lookup, and control tables. ................. 133
  Naming standard for tables ........................................ 133
  Table descriptions ........................................ 134
  Tables in the UNIX accounting subcomponent ........................................ 134
    XACCT_COMMAND_D_M .............. 135
    XACCT_DISK_D_M .................. 135
    XACCT_PRINT_D_M .............. 136
    XACCT_CONNECT_D_M .......... 136
    XACCT_DISKMV .................... 137
  Tables in the UNIX configuration subcomponent ........................................ 137
    XCONFIG_HW ..................... 137
    XCONFIG_SW ..................... 137
  Tables in the UNIX error subcomponent ........................................ 138
    XERROR_D_M ..................... 138
  Tables in the UNIX performance subcomponent ........................................ 138
    XPERF_CPU_H_D_M ............. 139
    XPERF_DISK_D_M .............. 140
    XPERF_DISKIO_H_D_M .......... 140
    XPERF_PAGING_H_D_M .......... 141
    XPERF_PS_INFO ............... 142
    XPERF_VM_INFO ............... 142
  Tables in the Windows component ........................................ 143
    WIN_PERF_DD_H_D_M .......... 143
    WIN_PERF_PU_H_D_M .......... 143
  Lookup tables ........................................ 144
    XACCT_CONNECT_TYPE ............. 144
    Example of table contents ........ 144
    X_NODE_NAME ..................... 145
    Example of table contents ........ 145
    XACCT_PRINT_TYPE ............. 145
    Example of table contents ........ 145
Chapter 13. Linux data tables ................. 147
  XCONF_HARDWARE ............... 147

Chapter 10. Installing and configuring the Windows component ........ 117
  Planning the implementation process ........................................ 117
  Considering which Windows subcomponents to install ........................................ 119
  Installing the Windows component on your Windows nodes ........................................ 119
    Step 1: Check Windows requirements ........................................ 119
    Disk space requirements ........................................ 119
    Software requirements ........................................ 120
    Step 2: Transfer Tivoli Decision Support for 
z/OS feature code to Windows ........................................ 120
  Capturing Windows component data and 
transmitting to OS/390 ........................................ 121
    Step 1: Customize Windows node ........................................ 121
    Parameter file values to change ........................................ 121
    Step 2: Set up scheduled tasks ........................................ 122
  Installing Windows component on the z/OS system ........................................ 123
  Collecting data under Tivoli Decision Support for 
z/OS ........................................ 125
  General collect job ........................................ 125
  Testing the installation ........................................ 125
  Putting the feature into production ........................................ 126

Emergency Handling and z/OS Recovery Management
<table>
<thead>
<tr>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notices</td>
</tr>
<tr>
<td>Trademarks</td>
</tr>
<tr>
<td>Bibliography</td>
</tr>
<tr>
<td>TDS publications</td>
</tr>
<tr>
<td>Glossary</td>
</tr>
<tr>
<td>Index</td>
</tr>
</tbody>
</table>

### Appendix. Support information 213

- Searching knowledge bases 213
- Searching the information center 213
- Searching the Internet 213
- Obtaining fixes 213
- Receiving weekly support updates 214
- Contacting IBM Software Support 214
  - Determining the business impact 215
  - Describing problems and gathering information 216
  - Submitting problems 216

### Notices

- Trademarks
- Bibliography
- Glossary
- Index
Figures

1. Organizing and presenting system performance data ........................................... 5
2. Implementation process for the Distributed component ........................................... 7
3. Tivoli Decision Support for z/OS
   Administration window ......................................................................................... 30
4. Components window ............................................................................................... 30
5. Installation Options window ................................................................................... 31
6. Daily steps involved in using Tivoli Decision Support for z/OS .............................. 32
7. Implementation process for the Windows component ............................................. 34
8. Tivoli Decision Support for z/OS
   Administration window ......................................................................................... 39
9. Components window ............................................................................................... 39
10. Installation Options window ............................................................................... 40
11. General COLLECT job .......................................................................................... 41
12. Daily steps involved in using Tivoli Decision Support for z/OS ......................... 42
13. General Distributed Systems Performance feature data flow ............................. 45
15. Example of Distrib. CPU Util. by System Daily Overview .................................... 56
17. Example of Distrib. Statistics by System, Daily Overview .................................... 58
18. Example of Distrib. File Systems, Daily Overview .............................................. 60
21. Example of Distrib. Page Space Util., Hourly Trend ........................................... 63
22. Example of Distributed CPU Utilization Exceptions ......................................... 64
23. Example of Windows Disk Usage for System, Hourly Trend ............................ 65
24. Example of Windows Disk Usage for Device, Daily Overview ............................ 66
25. Example of Windows CPU Utilization by System, Hourly Trend ....................... 67
26. Example of Windows CPU Utilization by System, Daily Overview .................. 68
27. Example of Windows Memory Usage by System, Hourly Trend ....................... 69
28. Example of Windows Memory Usage by System, Daily Overview .................... 70
29. Example of Windows System Overview Report ............................................... 71
30. Organizing and presenting system performance data ......................................... 82
31. Implementation process for the UNIX Performance component ....................... 86
32. Tivoli Decision Support for z/OS
   Administration window .......................................................................................... 103
33. Components window ............................................................................................. 103
34. Installation Options window ................................................................................... 104
35. General COLLECT job ........................................................................................... 108
36. Sample COLLECT job with an INCLUDE clause ................................................ 109
37. Daily steps involved in using Tivoli Decision Support for z/OS ......................... 110
38. General COLLECT job ........................................................................................... 117
39. Implementation process for the Windows component ......................................... 118
40. Tivoli Decision Support for z/OS
   Administration window .......................................................................................... 123
41. Components window ............................................................................................. 124
42. Installation Options window ................................................................................... 124
43. General COLLECT job ........................................................................................... 125
44. Daily steps involved in using Tivoli Decision Support for z/OS ......................... 126
45. General Distributed Systems Performance component data flow .................... 129
46. Example of UNIX Acct Commands by User, Daily Overview ........................... 153
47. Example of UNIX Acct Users by Command, Daily Overview .............................. 155
48. Example of UNIX Acct Cmd Resource Consumption, Daily Overview ............. 156
49. Example of UNIX Acct User Resource Usage, Monthly Overview .................... 157
50. Example of UNIX Acct Disk Blocks by User, Monthly Overview ....................... 159
51. Example of UNIX Acct Disk Blocks in 1000s, Monthly Trend ............................ 160
52. Example of UNIX Acct Users and Connects, Daily Overview ............................ 161
53. Example of UNIX Acct Printed Pages by User, Monthly Overview .................... 162
54. Example of UNIX Acct Printed Pages by System, Monthly Overview ............... 163
55. Example of UNIX Configuration of HW for a System, Overview ....................... 164
56. Example of UNIX Configuration of HW for Device Class, Overview................... 165
57. Example of UNIX Configuration of SW for a System, Overview ....................... 166
58. Example of UNIX Configuration of SW for Object, Overview ............................ 167
59. Example of UNIX Error by ID, Daily Overview ..................................................... 168
60. Example of UNIX Error by Type, Daily Overview ................................................. 169
61. Example of UNIX Error by Class, Daily Overview .............................................. 170
62. Example of UNIX Error by Resource, Daily Overview ........................................ 171
63. Example of UNIX Error by Resource, Monthly Trend .......................... 172
64. Example of UNIX Perf CPU Utilization by System, Hourly Trend .............. 173
65. Example of UNIX Perf CPU Utilization by System, Daily Overview .......... 174
67. Example of UNIX Perf Statistics all Systems, Daily Overview .............. 177
68. Example of UNIX Perf Vol Group and File Syst, Daily Overview ............ 178
69. Example of UNIX Perf Disk I/O for a Disk, Hourly Trend .................... 179
70. Example of UNIX Perf Disk I/O for System, Daily Overview .................. 180
71. Example of UNIX Perf Page Space Utilization, Hourly Trend ............... 182
72. Example of a LINUX - Percentile Work Size report .......................... 183
73. Example of a LINUX - Disk Space Allocation report .......................... 184
74. Example of a LINUX - Performance from User Memory report .......... 185
75. Example of a LINUX - Performance 'PS' for Volumes Info report ........ 186
76. Example of a LINUX - Hardware Configuration report ...................... 187
77. Example of a LINUX - Software Configuration report ...................... 188
78. Example of a LINUX - User Information report .............................. 189
79. Example of a LINUX - Process Information report ............................ 190
80. Example of a LINUX - Performance 'VM' for Swap Memory report .......... 191
81. Example of a LINUX - Performance 'VM' for CPU report .................... 192
82. Example of Windows Disk Usage for System, Hourly Trend .................. 193
83. Example of Windows Disk Usage for Device, Daily Overview ................ 194
84. Example of Windows CPU Utilization by System, Hourly Trend ............. 195
85. Example of Windows CPU Utilization by System, Daily Overview ........... 196
86. Example of Windows Memory Usage by System, Hourly Trend .............. 197
87. Example of Windows Memory Usage by System, Daily Overview ............ 198
88. Example of Windows System Overview Report ................................. 199
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 platforms, 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).
Tivoli Decision Support for z/OS library

- **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,
Typeface conventions

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
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 chapter topic 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 chapter 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 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.

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>
<thead>
<tr>
<th>Installation task</th>
<th>Disk space (approximate)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Load code in the file system</td>
<td>1MB</td>
</tr>
<tr>
<td>Create work files during the gathering process</td>
<td>5MB</td>
</tr>
</tbody>
</table>
Note: The 5MB of work files consists of:

Table 2. Details of gathering process disk space requirements

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

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: DRLxxx.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.
Installing and configuring your SLES nodes

**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
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 I28
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

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

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):
   - `drlmnntr`  
     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.
Installing and configuring your SLES nodes

**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

   ```bash
   #!/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

<table>
<thead>
<tr>
<th>Installation task</th>
<th>Disk space (approximate)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loading code in the file system</td>
<td>1MB</td>
</tr>
<tr>
<td>Creating work files during the gathering process</td>
<td>5MB</td>
</tr>
</tbody>
</table>

Note: The 5MB of work files consists of:

Table 5. Details of gathering process disk space requirements

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

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

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.
   ```
Installing and configuring your RHEL nodes

*dr1perf*
The process responsible for collecting disk I/O, memory usage, user and process information.

*dr1disk*
The process responsible for collecting disk usage information.

*dr1page*
The process responsible for collecting paging space information.

*dr1xfer*
The process responsible for transferring all information to the host.

*dr1clnp*
The process responsible for cleaning up old files.

3. Check the nph 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 dr1perf, dr1disk and dr1page processes). It may take some time for all of these to appear; 10 minutes in the default configuration.
5. Ensure that the dr1xfer 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 dr1clnp process successfully clears files from the node. This can be tested by creating a log.dr1perf-1900-01-01 in the var directory, then executing the command dr1ctrl 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 dr1user using an editor. The contents of this file should be:

   ```bash
   #!/bin/sh
   su - dr1user "/home/drluser/drl/bin/rc.drluser"
   ```
3. Change the permissions of the startup file with chmod 755 dr1user.
4. Create a symbolic link for the startup file:

   ```bash
   cd /etc/rc5.d
   ln -s /etc/init.d/drluser 599drluser
   ```
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.
Installing and configuring 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

<table>
<thead>
<tr>
<th>Installation task</th>
<th>Disk space (approximate)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loading code in the file system</td>
<td>1MB</td>
</tr>
<tr>
<td>Creating work files during the gathering process</td>
<td>5MB</td>
</tr>
</tbody>
</table>

Note: The 5MB of work files consists of:

Table 8. Details of gathering process disk space requirements

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

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: DRLxxx.SDRLW$ (DRLDIST)

Download and install this the tar.gz file using these steps:
Installing and configuring your HP-UX nodes

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**

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

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):
   - drlmontr
     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>
<thead>
<tr>
<th>Table 10. Disk space requirements on each distributed node</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installation task</td>
</tr>
<tr>
<td>Loading code in the file system</td>
</tr>
<tr>
<td>Creating work files during the gathering process</td>
</tr>
</tbody>
</table>

**Note:** The 5MB of work files consists of:

<table>
<thead>
<tr>
<th>Table 11. Details of gathering process disk space requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Record type</td>
</tr>
<tr>
<td>File systems (DISKFSA)</td>
</tr>
<tr>
<td>Paging spaces (PERFPSA)</td>
</tr>
<tr>
<td>Processor and memory information (PERFVMA)</td>
</tr>
<tr>
<td>Disk I/O information (PERFIOA)</td>
</tr>
<tr>
<td>Auxiliary amounts</td>
</tr>
</tbody>
</table>

**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: DRLxxx.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.

tsoi d
   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): tsoi d
   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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>General:SysName</td>
<td>The system name for identification purposes in Tivoli Decision Support for z/OS. This should be unique across all distributed nodes.</td>
</tr>
<tr>
<td>Xfer:Host</td>
<td>The host name or IP address of the host to transmit the data to.</td>
</tr>
<tr>
<td>Xfer:User</td>
<td>The TSO user ID at the host.</td>
</tr>
<tr>
<td>Xfer:Password</td>
<td>The password for that TSO user ID at the host.</td>
</tr>
<tr>
<td>Perf:IoCmd</td>
<td>The prefabricated command indicator for collecting disk I/O statistics. Should be @0002 for AIX.</td>
</tr>
<tr>
<td>Perf:VmCmd</td>
<td>The prefabricated command indicator for collecting memory statistics. Should be @0002 for AIX.</td>
</tr>
<tr>
<td>Perf:UserCmd</td>
<td>The prefabricated command indicator for collecting user statistics. Should be @0000 for AIX.</td>
</tr>
<tr>
<td>Perf:ProcCmd</td>
<td>The prefabricated command indicator for collecting process statistics. Should be @0000 for AIX.</td>
</tr>
<tr>
<td>Disk:Fscmd</td>
<td>The prefabricated command indicator for collecting file system statistics. Should be @0001 for AIX.</td>
</tr>
<tr>
<td>Page:PsCmd</td>
<td>The prefabricated command indicator for collecting paging space statistics. Should be @0002 for AIX.</td>
</tr>
</tbody>
</table>
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):
   - drlmmtr
     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:
   ```sh
   #!/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
   ```sh
   cd /etc/rc.d/rc2.d
   ln -s /etc/rc.d/init.d/drluser 999drluser
   ```
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>
<thead>
<tr>
<th>Installation task</th>
<th>Disk space (approximate)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loading code in the file system</td>
<td>1MB</td>
</tr>
<tr>
<td>Creating work files during the gathering process</td>
<td>5MB</td>
</tr>
</tbody>
</table>

Note: The 5MB of work files consists of:

Table 14. Details of gathering process disk space requirements

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>General:SysName</td>
<td>The system name for identification purposes in Tivoli Decision Support for z/OS. This should be unique across all distributed nodes.</td>
</tr>
</tbody>
</table>
Installing and configuring your Solaris nodes

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

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

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 dr1user user, start up the gathering processes with the command dr1ctrl start.

2. Ensure that the following processes are running (assuming you haven’t disabled them by changing General:RunProcs parameter in the configuration file):
   - dr1mtr, the monitor process responsible for keeping everything running.
   - dr1perf, the process responsible for collecting disk I/O, memory usage, user and process information.
   - dr1disk, the process responsible for collecting disk usage information.
   - dr1page, the process responsible for collecting paging space information.
   - dr1xfer, the process responsible for transferring all information to the host.
   - dr1clnp, 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 dr1perf, dr1disk and dr1page processes). Note that it may take some time for all of these to appear, ten minutes in the default configuration.

5. Ensure that the dr1xfer 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 dr1clnp 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 dr1ctrl 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:
   ```sh
   #!/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:
   ```sh
   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.
Installing Distributed component on the z/OS system

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 4).

Figure 3. Tivoli Decision Support for z/OS Administration window

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).
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.

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.
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>
<thead>
<tr>
<th>Installation task</th>
<th>Disk space (approximate)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loading code in the file system. For example, c:\tdszwin\bin:</td>
<td>1MB</td>
</tr>
<tr>
<td>Creating configuration file in. For example, c:\tdszwin\etc:</td>
<td>1MB</td>
</tr>
<tr>
<td>Creating directory for data files in. For example, c:\tdszwin\var:</td>
<td>10MB (see note)</td>
</tr>
</tbody>
</table>

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 DRLxxx.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 'drll81.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:\tdszw\etc\drl.cfg` to change the default values supplied by the Windows component to those for your installation.

Parameter file values to change

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>NodeName</td>
<td>IBM-BOX01</td>
</tr>
<tr>
<td>XmitServer</td>
<td>pthbox3.au.ibm.com</td>
</tr>
<tr>
<td>XmitUser</td>
<td>drluser</td>
</tr>
<tr>
<td>XmitPassword</td>
<td>drlpwd</td>
</tr>
<tr>
<td>DrlPerfdXlat</td>
<td>'drldata.perfd.n{n5,5}.d{f12,6}'</td>
</tr>
</tbody>
</table>

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 `drlperf2_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.
Installing the Windows component on your Windows nodes

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

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).
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 DRLxxx.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 
/*/   DRLOG. 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.

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

![Data flow diagram](image)

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

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

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 chapter topic 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”
- “DIST_DISK_D, _M” on page 49
- “DIST_DISKIO_H, _D” on page 49
- “DIST_PAGING_H, _D, _M” on page 50

**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

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

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

**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

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

**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
### Tables in the Distributed component

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

**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

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

<table>
<thead>
<tr>
<th>Column</th>
<th>Key</th>
<th>Datatype</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>K</td>
<td>DATE</td>
<td>The date of the record. For the _M table, dates are stored as the first day of the month.</td>
</tr>
<tr>
<td>TIME</td>
<td>K</td>
<td>TIME</td>
<td>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.</td>
</tr>
<tr>
<td>NODE_NAME</td>
<td>K</td>
<td>CHAR(16)</td>
<td>The node name of the machine.</td>
</tr>
<tr>
<td>DEVICE_NAME</td>
<td>K</td>
<td>CHAR(8)</td>
<td>The device name for the device.</td>
</tr>
<tr>
<td>FREE_SPACE_AVG</td>
<td></td>
<td>FLOAT</td>
<td>Averages the free space on the device, based on SAMPLES.</td>
</tr>
<tr>
<td>FREE_SPACE_MAX</td>
<td></td>
<td>FLOAT</td>
<td>The maximum free space available.</td>
</tr>
<tr>
<td>FREE_SPACE_MIN</td>
<td></td>
<td>FLOAT</td>
<td>The minimum free space available.</td>
</tr>
<tr>
<td>PERCENT_FREE_AVG</td>
<td></td>
<td>FLOAT</td>
<td>Averages the free space percentage on the device, based on SAMPLES.</td>
</tr>
<tr>
<td>PERCENT_FREE_MAX</td>
<td></td>
<td>FLOAT</td>
<td>The maximum free space percentage.</td>
</tr>
<tr>
<td>PERCENT_FREE_MIN</td>
<td></td>
<td>FLOAT</td>
<td>The minimum free space percentage.</td>
</tr>
<tr>
<td>SAMPLES</td>
<td></td>
<td>INTEGER</td>
<td>Count of the records consolidated into a table row. Used for working out averages.</td>
</tr>
<tr>
<td>TOTALDEVICE_SIZE</td>
<td></td>
<td>FLOAT</td>
<td>Device size at the end of the period.</td>
</tr>
</tbody>
</table>

#### 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

<table>
<thead>
<tr>
<th>Column</th>
<th>Key</th>
<th>Datatype</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>K</td>
<td>DATE</td>
<td>The date of the record. For the _M table, dates are forced back to the first day of the month.</td>
</tr>
</tbody>
</table>
### Tables in the Windows component

<table>
<thead>
<tr>
<th>Column</th>
<th>Key</th>
<th>Datatype</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIME</td>
<td>K</td>
<td>TIME</td>
<td>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.</td>
</tr>
<tr>
<td>NODE_NAME</td>
<td>K</td>
<td>CHAR(16)</td>
<td>The node name of the machine.</td>
</tr>
<tr>
<td>CPU_COUNT</td>
<td></td>
<td>SMALLINT</td>
<td>Number of CPUs and/or cores.</td>
</tr>
<tr>
<td>CPU_USAGE_AVG</td>
<td></td>
<td>FLOAT</td>
<td>Averages the average CPU usage. This figure is averaged based on SAMPLES.</td>
</tr>
<tr>
<td>CPU_USAGE_MAX</td>
<td></td>
<td>FLOAT</td>
<td>The maximum CPU usage.</td>
</tr>
<tr>
<td>CPU_USAGE_MIN</td>
<td></td>
<td>FLOAT</td>
<td>The minimum CPU usage.</td>
</tr>
<tr>
<td>MEMORY_FREE_AVG</td>
<td></td>
<td>FLOAT</td>
<td>Averages the average free memory. This figure is averaged based on SAMPLES.</td>
</tr>
<tr>
<td>MEMORY_FREE_MAX</td>
<td></td>
<td>FLOAT</td>
<td>The maximum free memory.</td>
</tr>
<tr>
<td>MEMORY_FREE_MIN</td>
<td></td>
<td>FLOAT</td>
<td>The minimum free memory.</td>
</tr>
<tr>
<td>MEMORY_SIZE</td>
<td></td>
<td>FLOAT</td>
<td>The total memory size at the end of the period.</td>
</tr>
<tr>
<td>SAMPLES</td>
<td></td>
<td>INTEGER</td>
<td>Count of the records consolidated into a table row. Used for working out actuals.</td>
</tr>
</tbody>
</table>


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 chapter 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 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:

\[
\text{component sequence}
\]

where:

\begin{itemize}
  \item \text{component}
    \begin{itemize}
      \item Identifies the component to which the report belongs. It can be one of the following:
      \begin{itemize}
        \item \text{DIST} for the Distributed component.
        \item \text{WIN} for the Windows component.
      \end{itemize}
    \end{itemize}
  \item \text{sequence}
    \begin{itemize}
      \item A variable-length sequential number identifying the report.
    \end{itemize}
\end{itemize}

Examples of report IDs are:
- \text{DIST04}
- \text{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.
Report format and general description

Source
Each report contains information from one or more source tables. The report descriptions in this chapter 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.”
- "Distrib. CPU Util. by System, Daily Overview” on page 55.
- "Distrib. Disk I/O for Disk, Hourly Trend” on page 60.
- "Distrib. CPU Util. Exceptions” on page 63.

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:
Distributed reports

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

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.

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

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

---

![Distrib. CPU Util. by System, Daily Overview](chart)

**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

<table>
<thead>
<tr>
<th>Hour</th>
<th>CPU usage avg %</th>
<th>CPU usage max %</th>
<th>Mem free KB avg</th>
<th>Mem free KB min</th>
<th>Paging rate avg</th>
<th>Paging rate max</th>
<th>Procs avg</th>
<th>Procs max</th>
<th>Users avg</th>
<th>Users max</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3.1</td>
<td>5</td>
<td>399103</td>
<td>398708</td>
<td>0.00</td>
<td>0</td>
<td>161.2</td>
<td>192</td>
<td>2.3</td>
<td>4</td>
</tr>
<tr>
<td>1</td>
<td>11.3</td>
<td>32</td>
<td>399103</td>
<td>398708</td>
<td>0.00</td>
<td>0</td>
<td>122.3</td>
<td>144</td>
<td>7.1</td>
<td>14</td>
</tr>
<tr>
<td>2</td>
<td>8.5</td>
<td>17</td>
<td>399103</td>
<td>398708</td>
<td>0.00</td>
<td>0</td>
<td>157.5</td>
<td>166</td>
<td>1.4</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>7.7</td>
<td>23</td>
<td>399103</td>
<td>398708</td>
<td>0.00</td>
<td>0</td>
<td>124.7</td>
<td>161</td>
<td>8.1</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
</tbody>
</table>

| 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         |

---

**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.
Distributed reports

- **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

<table>
<thead>
<tr>
<th>Node name</th>
<th>CPU usage avg</th>
<th>CPU usage max</th>
<th>Mem free KB avg</th>
<th>Mem free KB max</th>
<th>Paging rate avg</th>
<th>Paging rate max</th>
<th>Processes avg</th>
<th>Processes max</th>
<th>Users avg</th>
<th>Users max</th>
</tr>
</thead>
<tbody>
<tr>
<td>xub11a</td>
<td>6.2</td>
<td>41</td>
<td>399103</td>
<td>398708</td>
<td>0.00</td>
<td>0</td>
<td>141.9</td>
<td>172</td>
<td>17.4</td>
<td>30</td>
</tr>
<tr>
<td>xub11b</td>
<td>21.7</td>
<td>31</td>
<td>399103</td>
<td>398708</td>
<td>0.00</td>
<td>0</td>
<td>282.8</td>
<td>344</td>
<td>21.3</td>
<td>48</td>
</tr>
<tr>
<td>xub11c</td>
<td>9.3</td>
<td>12</td>
<td>399103</td>
<td>398708</td>
<td>0.00</td>
<td>0</td>
<td>223.7</td>
<td>316</td>
<td>16.5</td>
<td>36</td>
</tr>
<tr>
<td>xub11d</td>
<td>37.2</td>
<td>63</td>
<td>399103</td>
<td>398708</td>
<td>0.00</td>
<td>0</td>
<td>164.6</td>
<td>238</td>
<td>28.0</td>
<td>52</td>
</tr>
<tr>
<td>xub11e</td>
<td>3.1</td>
<td>15</td>
<td>399103</td>
<td>398708</td>
<td>0.00</td>
<td>0</td>
<td>105.5</td>
<td>200</td>
<td>19.6</td>
<td>36</td>
</tr>
<tr>
<td>xub11f</td>
<td>12.4</td>
<td>38</td>
<td>399103</td>
<td>398708</td>
<td>0.00</td>
<td>0</td>
<td>246.4</td>
<td>322</td>
<td>24.1</td>
<td>44</td>
</tr>
</tbody>
</table>

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
Distributed reports

<table>
<thead>
<tr>
<th>Device</th>
<th>File system</th>
<th>Space size avg (MB)</th>
<th>Space free avg (MB)</th>
<th>Space used avg (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>/dev/sda1</td>
<td>/</td>
<td>10000000.0</td>
<td>2470000.0</td>
<td>75.3</td>
</tr>
<tr>
<td>none</td>
<td>/dev</td>
<td>505348.0</td>
<td>504732.0</td>
<td>1.0</td>
</tr>
<tr>
<td>none</td>
<td>/dev/shm</td>
<td>512388.0</td>
<td>512376.0</td>
<td>1.0</td>
</tr>
<tr>
<td>none</td>
<td>/var/lock</td>
<td>512388.0</td>
<td>512388.0</td>
<td>0.0</td>
</tr>
<tr>
<td>none</td>
<td>/var/run</td>
<td>512388.0</td>
<td>512284.0</td>
<td>1.0</td>
</tr>
</tbody>
</table>

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
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 19 on page 61) 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
Distributed reports

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

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

<table>
<thead>
<tr>
<th>Hour</th>
<th>Page space name</th>
<th>Disk name</th>
<th>Page space size (MB)</th>
<th>Page space used (MB)</th>
<th>Page space used (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>/dev/sda15</td>
<td>/dev/sda15</td>
<td>2048</td>
<td>0</td>
<td>0.0</td>
</tr>
<tr>
<td>1</td>
<td>/dev/sda15</td>
<td>/dev/sda15</td>
<td>2048</td>
<td>1024</td>
<td>50.0</td>
</tr>
<tr>
<td>2</td>
<td>/dev/sda15</td>
<td>/dev/sda15</td>
<td>2048</td>
<td>512</td>
<td>25.0</td>
</tr>
<tr>
<td>22</td>
<td>/dev/sda15</td>
<td>/dev/sda15</td>
<td>2048</td>
<td>1536</td>
<td>75.0</td>
</tr>
<tr>
<td>23</td>
<td>/dev/sda15</td>
<td>/dev/sda15</td>
<td>2048</td>
<td>256</td>
<td>12.5</td>
</tr>
</tbody>
</table>

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
Distributed reports

Report group
Distributed

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

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

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

<table>
<thead>
<tr>
<th>Node name</th>
<th>Type</th>
<th>Date</th>
<th>Hour</th>
<th>Average CPU</th>
<th>Maximum CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>xub11a</td>
<td>HOURLY</td>
<td>2012-05-16</td>
<td>6</td>
<td>6.0</td>
<td>41</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>7</td>
<td>6.0</td>
<td>71</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>11</td>
<td>4.0</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2012-05-17</td>
<td>6</td>
<td>6.3</td>
<td>41</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>11</td>
<td>14.0</td>
<td>55</td>
</tr>
<tr>
<td>xub11b</td>
<td>HOURLY</td>
<td>2012-05-16</td>
<td>7</td>
<td>12.1</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>11</td>
<td>11.6</td>
<td>49</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2012-05-17</td>
<td>11</td>
<td>4.0</td>
<td>52</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>12</td>
<td>4.0</td>
<td>51</td>
</tr>
</tbody>
</table>

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 exceed the relevant threshold.

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

Reports in the Windows component

This section describes the Windows component reports:

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'

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

---

**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.
Windows component reports

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**

<table>
<thead>
<tr>
<th>DATE</th>
<th>NODE_NAME</th>
<th>DEVICE</th>
<th>FREE SPACE MIN (MB)</th>
<th>FREE SPACE AVG (MB)</th>
<th>FREE SPACE MAX (MB)</th>
<th>PERCENT FREE MIN</th>
<th>PERCENT FREE AVG</th>
<th>PERCENT FREE MAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>2009-06-12</td>
<td>IBM-PAXPRIM</td>
<td>C:</td>
<td>61565</td>
<td>61652</td>
<td>61720</td>
<td>67</td>
<td>67</td>
<td>67</td>
</tr>
</tbody>
</table>

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, 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

---

<table>
<thead>
<tr>
<th>NODE NAME</th>
<th>DATE</th>
<th>TIME</th>
<th>USAGE</th>
<th>PERCENT</th>
<th>USAGE</th>
<th>PERCENT</th>
<th>USAGE</th>
<th>PERCENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM-PAXPRIM</td>
<td>2009-06-12</td>
<td>00.00.00</td>
<td>4</td>
<td>4</td>
<td>7</td>
<td>1</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>01.00.00</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>02.00.00</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>03.00.00</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>04.00.00</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>05.00.00</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>1</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>06.00.00</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>07.00.00</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>08.00.00</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>09.00.00</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>1</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10.00.00</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>11.00.00</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>12.00.00</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>13.00.00</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>14.00.00</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>15.00.00</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>16.00.00</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>17.00.00</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>1</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>18.00.00</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>1</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>19.00.00</td>
<td>6</td>
<td>7</td>
<td>7</td>
<td>1</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>20.00.00</td>
<td>7</td>
<td>7</td>
<td>8</td>
<td>1</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>21.00.00</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>1</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>22.00.00</td>
<td>8</td>
<td>9</td>
<td>9</td>
<td>1</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>23.00.00</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>1</td>
<td>9</td>
<td>9</td>
</tr>
</tbody>
</table>

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

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

Windows component reports
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

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

<table>
<thead>
<tr>
<th>NODE NAME</th>
<th>DATE</th>
<th>MEMORY (KB)</th>
<th>MEMORY (KB)</th>
<th>MEMORY (KB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM-PAXPRIM</td>
<td>2009-06-12</td>
<td>893760</td>
<td>1159459</td>
<td>1408564</td>
</tr>
</tbody>
</table>

Tivoli Decision Support: WIN006

---

**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

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

**Date:** '2009-04-15' to '2009-04-15'

<table>
<thead>
<tr>
<th>NODE NAME</th>
<th>CPU USAGE MIN</th>
<th>CPU USAGE AVG</th>
<th>CPU USAGE MAX</th>
<th>MEMORY FREE MIN (KB)</th>
<th>MEMORY FREE AVG (KB)</th>
<th>MEMORY FREE MAX (KB)</th>
<th>FREE SPACE MIN (MB)</th>
<th>FREE SPACE AVG (MB)</th>
<th>FREE SPACE MAX (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM-PAXWXK3</td>
<td>2009-04-15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 29. Example of Windows System Overview Report

Tivoli Decision Support: WIN007

Chapter 6. Reports 71
Chapter 7. Log record definitions

This chapter describes the following record definitions:

- “DIST_PERF_VM_A” on page 74
- “DIST_DISK_FS_A” on page 74
- “DIST_PERF_IO_A” on page 74
- “DIST_PERF_PS_A” on page 75
- “WIN_PERF_DD” on page 76
- “WIN_PERF_PU” on page 76

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

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

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

### Main section

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

#### Record section DEVICE

**Offset** 62  
**Length** Length of DEVLEN + 4

<table>
<thead>
<tr>
<th>Field name</th>
<th>Offset</th>
<th>Length</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEVLEN</td>
<td>0</td>
<td>4</td>
<td>EXTERNAL INTEGER</td>
<td>Length of field DEVNAME</td>
</tr>
<tr>
<td>DEVNAME</td>
<td>4</td>
<td>*</td>
<td>CHAR(*)</td>
<td>Device name</td>
</tr>
</tbody>
</table>

#### Record section SPACE

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

<table>
<thead>
<tr>
<th>Field name</th>
<th>Offset</th>
<th>Length</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FSLEN</td>
<td>0</td>
<td>4</td>
<td>EXTERNAL INTEGER</td>
<td>Length of section SPACE_NAME</td>
</tr>
<tr>
<td>FSNAME</td>
<td>4</td>
<td>*</td>
<td>CHAR(*)</td>
<td>Space name</td>
</tr>
</tbody>
</table>

DIST_PERF_IO_A

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

### Main section

<table>
<thead>
<tr>
<th>Field name</th>
<th>Offset</th>
<th>Length</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RECLEN</td>
<td>0</td>
<td>2</td>
<td>BINARY</td>
<td>Record length</td>
</tr>
</tbody>
</table>
### DIST_PERF_PS_A

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

#### Main section

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

```wmi
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

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

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:

```wmi
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

<table>
<thead>
<tr>
<th>Column</th>
<th>Position</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>0</td>
<td>2</td>
<td>PU to indicate this is a performance usage record.</td>
</tr>
<tr>
<td>Date</td>
<td>3</td>
<td>8</td>
<td>YYYYMMDD of the record.</td>
</tr>
<tr>
<td>Time</td>
<td>12</td>
<td>6</td>
<td>HHMMSS of the record.</td>
</tr>
<tr>
<td>Node name</td>
<td>19</td>
<td>16</td>
<td>Node name of the machine.</td>
</tr>
<tr>
<td>Column</td>
<td>Position</td>
<td>Size</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>----------</td>
<td>------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CPU count</td>
<td>36</td>
<td>2</td>
<td>Number of CPUs in this system.</td>
</tr>
<tr>
<td>Average CPU usage</td>
<td>39</td>
<td>3</td>
<td>Zero-padded average CPU percentage usage for period. This is the average of all the snapshots in the period.</td>
</tr>
<tr>
<td>Minimum CPU usage</td>
<td>43</td>
<td>3</td>
<td>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 <code>xb_XPerfCpuRun</code> configuration item. This is the minimum snapshot for the period.</td>
</tr>
<tr>
<td>Maximum CPU usage</td>
<td>47</td>
<td>3</td>
<td>Zero-padded maximum CPU percentage usage for period. This is the minimum snapshot for the period.</td>
</tr>
<tr>
<td>Memory size</td>
<td>51</td>
<td>9</td>
<td>Zero-padded total memory size at end of period, in kilobytes.</td>
</tr>
<tr>
<td>Average memory</td>
<td>61</td>
<td>9</td>
<td>Zero-padded average memory free for period, in kilobytes.</td>
</tr>
<tr>
<td>Minimum memory</td>
<td>71</td>
<td>9</td>
<td>Zero-padded minimum memory free for period, in kilobytes (1024 bytes).</td>
</tr>
<tr>
<td>Maximum memory</td>
<td>81</td>
<td>9</td>
<td>Zero-padded maximum memory free for period, in kilobytes.</td>
</tr>
</tbody>
</table>

Chapter 7. Log record definitions
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 where AIX, 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**

<table>
<thead>
<tr>
<th>Description of created log files</th>
</tr>
</thead>
</table>

**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.
Gathering and collecting performance 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 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
Gathering and collecting performance data

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)
Gathering and collecting performance data
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.
Planning the implementation process

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

![Diagram](image)

**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 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

<table>
<thead>
<tr>
<th>Installation task</th>
<th>Disk space (approximate)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loading code in the file system /usr/lpp:</td>
<td>400KB</td>
</tr>
<tr>
<td>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:</td>
<td>43MB for 10 users (see table note)</td>
</tr>
</tbody>
</table>

Note: The 43MB consists of:

Component  
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.
Installing the UNIX Performance component on your UNIX nodes

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**
  
  DRLxxx.SDRLWS(DRLAIX)
- **HP-UX**
  
  DRLxxx.SDRLWS(DRLHP11)
- **Sun Solaris**
  
  DRLxxx.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.

   ```shell
   # 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)' epdmpcp.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 epdmxxxx 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

<table>
<thead>
<tr>
<th>Directory</th>
<th>Description</th>
<th>Created files:</th>
</tr>
</thead>
</table>
| /usr/lpp/epdm/bin/ | This directory was created when the `tar` command was run. It contains executable UNIX Performance component commands. | drlpacct
drwtmp
drldtmp
dlrqacct (see Note 1)
dlerrprt (see Note 1)
дрюконфhw (see Note 1)
дрюконфsw (see Note 1)
drlgather
drmlinks
drldtmp
drlacct
drlrmlinks
drlstat
drvmstat
drlnfs
drllrmpf
drlsend (see Note 2)
drlactperf (see Note 2)
drlactperf (see Note 2)
drldeactperf (see Note 2)
drlrlday
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. | drlcr
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:
Installing the UNIX Performance component on your UNIX nodes

- 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 (hmmn)
   c. End of prime time (hmmn)

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/suadm -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.pqueue
```

```
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 cron 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:

```
/prvar/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 > /var/adm/acct/nite/opacct
```

Set the file permissions and ownership:

```
chmod 550 runacct.local
chown adm:adm runacct.local
```
Installing the UNIX Performance component on your UNIX nodes

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
Installing the UNIX Performance component on your UNIX nodes

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 be also 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**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>logPath</td>
<td>The name of directory where all the data log files will be stored.</td>
</tr>
<tr>
<td>exePath</td>
<td>The name of the directory where all commands are.</td>
</tr>
<tr>
<td>msgCatalog</td>
<td>The name of the file where all messages are stored.</td>
</tr>
<tr>
<td>epdmHost</td>
<td>Host name (TCP nodename or TCP internet address) of the OS/390 host to receive the UNIX Performance component data files.</td>
</tr>
<tr>
<td>epdmPrefix</td>
<td>Log data set prefix, which you set according to your own requirements.</td>
</tr>
<tr>
<td>minFree</td>
<td>Minimum free disk space in KB on <code>/var/epdm</code> before running the daily <code>drlgather</code>.</td>
</tr>
<tr>
<td>site</td>
<td>OS/390 data set characteristics for the data files sent using TCP/IP FTP to OS/390.</td>
</tr>
<tr>
<td>maxRetries</td>
<td>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 <code>drlgather</code> will repeat attempt the file transfer of the data files for a number of times given by the parameter 'maxRetries'. Using</td>
</tr>
</tbody>
</table>
Installing the UNIX Performance component on your UNIX nodes

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:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>perfInterval</td>
<td>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.</td>
</tr>
</tbody>
</table>
| 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.dyyymmdd.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.dyyymmdd.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.
Installing the UNIX Performance component on your UNIX nodes

Table 19. Creating a directory from the drlcrfs command

<table>
<thead>
<tr>
<th>Directory</th>
<th>Description</th>
<th>Files to be created when UNIX Performance component running</th>
</tr>
</thead>
<tbody>
<tr>
<td>/var/epdm/</td>
<td>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.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>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</td>
<td>sk log data files: (see Notes 2, 3, 4) node_name.pacct.d{yymmdd}.1 node_name.wtmp.d{yymmdd}.1 node_name.dtpm.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)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>sk message log files: (see Notes 2, 3) node_name.drlperfd.errorlog node_name.log.d{yymmdd} node_name.errorlog.d{yymmdd}</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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
   \textit{“drlpsperfd” on page 102}

\textbf{Results}

Performance data will now be entered in the following files:

- \textbf{work files}
  - \textit{node\_name.perf.work}
  - \textit{node\_name.disk.work}
  - \textit{node\_name.perfio.work}
  - \textit{node\_name.perfvm.work}

- \textbf{log data files}
  - \textit{node\_name.perf.dyyymmdd}
  - \textit{node\_name.disk.dyyymmdd}

The file \textit{node\_name.drlperfd.errorlog} contains possible error messages. This file
remains empty providing the drlactperfd command runs successfully.

The file \textit{node\_name.drlperfd.log} contains information messages which inform you,
for example, of the files that have been transferred to OS/390.

\textbf{Step 6: Invoke drlperfd at start time}

\textbf{About this task}

\textit{(performance component only)}

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

\textbf{Procedure}

1. For AIX, enter the command \texttt{mkitab epdm:2:wait:/etc/rc.epdmunix}. For HP
   and Sun operating systems, edit \texttt{/etc/inittab} and add the epdm line at the end
   as shown.
   
   \begin{verbatim}
   # 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
   \end{verbatim}

2. Copy \texttt{rc.epdmunix} to \texttt{/etc/} using the command \texttt{cp /usr/lpp/epdm/etc/}
   \texttt{rc.epdmunix /etc/}.

\textbf{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 \textit{“Step 6: Invoke drlperfd at
start time.”} The program that starts the four processes is \texttt{drlperfd}, which
periodically logs:
Installing the UNIX Performance component on your UNIX nodes

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

**Note:** An explanation of node_name, dyymmdd, 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.

<table>
<thead>
<tr>
<th>Time</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>01:00</td>
<td>UNIX starts the dodisk command via cron.</td>
</tr>
<tr>
<td>01:30</td>
<td>UNIX starts the runacct command via cron.</td>
</tr>
<tr>
<td>02:00</td>
<td>UNIX starts the drlgather command via cron.</td>
</tr>
</tbody>
</table>

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.
- dlrrerrpt, 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.

<table>
<thead>
<tr>
<th>Time</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>02:30</td>
<td>UNIX starts the monacct command via cron daemon.</td>
</tr>
</tbody>
</table>

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).
Operating the UNIX Performance component on UNIX

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.dymmdd, 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 777 Nov 24 19:12 dr1.cfg
-rw-r--r-- 1 root system 7847 Feb 07 10:43 nodel.disk.work
-rw-r--r-- 1 root system 0 Jan 02 19:25 nodel.dr1perf.errlog
-rw-r--r-- 1 root system 618581 Jan 22 02:11 nodel.epdmunix.d950127.1
-rw-r--r-- 1 root system 93 Jan 22 02:12 nodel.perfwork.d950127
-rw-r--r-- 1 root system 436 Jan 22 02:12 nodel.permsg.d950127
-rw-r--r-- 1 root system 224 Jan 22 02:12 nodel.pererr.d950127
-rw-r--r-- 1 root system 0 Jan 22 02:00 nodel.errorlog.d950122
-rw-r--r-- 1 root system 0 Jan 23 02:00 nodel.errorlog.d950123
-rw-r--r-- 1 root system 0 Jan 24 02:00 nodel.errorlog.d950124
-rw-r--r-- 1 root system 0 Jan 25 02:00 nodel.errorlog.d950125
-rw-r--r-- 1 root system 0 Jan 26 02:00 nodel.errorlog.d950126
-rw-r--r-- 1 root system 0 Jan 27 02:00 nodel.errorlog.d950127
-rw-r--r-- 1 root system 0 Jan 28 02:00 nodel.errorlog.d950128
-rw-r--r-- 1 root system 0 Jan 29 02:00 nodel.errorlog.d950129
-rw-r--r-- 1 root system 0 Jan 22 02:00 nodel.errorlog.d950122
-rw-r--r-- 1 root system 168 Feb 01 02:01 nodel.errorlog.d950201
-rw-r--r-- 1 root system 0 Feb 02 02:00 nodel.errorlog.d950202
-rw-r--r-- 1 root system 0 Feb 03 02:00 nodel.errorlog.d950203
-rw-r--r-- 1 root system 11 Feb 07 19:10 nodel.errpt.time
-rw-r--r-- 1 root system 2307 Jan 22 02:02 nodel.log.d950122
-rw-r--r-- 1 root system 2321 Jan 22 02:02 nodel.log.d950123
-rw-r--r-- 1 root system 2302 Jan 24 02:03 nodel.log.d950124
-rw-r--r-- 1 root system 2409 Jan 25 02:04 nodel.log.d950125
-rw-r--r-- 1 root system 2043 Jan 26 02:02 nodel.log.d950126
-rw-r--r-- 1 root system 2312 Jan 27 02:03 nodel.log.d950127
-rw-r--r-- 1 root system 2351 Jan 28 02:03 nodel.log.d950128
-rw-r--r-- 1 root system 2234 Jan 29 02:02 nodel.log.d950129
-rw-r--r-- 1 root system 2287 Jan 30 02:04 nodel.log.d950130
-rw-r--r-- 1 root system 2357 Jan 31 02:03 nodel.log.d950131
-rw-r--r-- 1 root system 2308 Feb 01 02:02 nodel.log.d950201
-rw-r--r-- 1 root system 2319 Feb 02 02:04 nodel.log.d950202
-rw-r--r-- 1 root system 2401 Feb 03 02:02 nodel.log.d950203
-rw-r--r-- 1 root system 35293 Feb 07 11:06 nodel.perf.work
-rw-r--r-- 1 root system 111 Feb 07 11:07 nodel.perfio.work
-rw-r--r-- 1 root system 116 Feb 07 11:07 nodel.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 nodel.fpwkwork.d950127, nodel.fptmsg.d950127, and nodel.fpterr.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 (nodel.epdmunix.d950127.1), using the command dr1send.

You can also see that an error has occurred on February 01, because the file nodel.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:

```
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.

**drideactperfd**
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 retryInterval 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.

   ![Figure 32. Tivoli Decision Support for z/OS Administration window](image)

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

   ![Figure 33. Components window](image)

   ![Other Utilities Help](image)

   From the Components window, select the components to install (here, the AIX accounting component) and press F6.

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).
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.Dyymmdd

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.EPDMAIN

USER1.ADAM.EPDMUNIX.D921214 G0001V00
USER1.ADAM.EPDMUNIX.D921215 G0002V00
Collecting data under Tivoli Decision Support for z/OS

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 G0007V00, 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.
Collecting data under Tivoli Decision Support for z/OS

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.
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.
Putting the feature into production

Figure 37 shows the 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:
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 epdmLINUX 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)
Considering which Linux subcomponents to install

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:

   ```plaintext
   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**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>logPath</td>
<td><code>/var/epdm</code></td>
</tr>
<tr>
<td>exePath</td>
<td><code>/usr/lpp/epdm/bin</code> (Do not change)</td>
</tr>
<tr>
<td>epdmHost</td>
<td><code>yourmvs</code></td>
</tr>
<tr>
<td>epdmPrefix</td>
<td><code>EPDMLINUX</code></td>
</tr>
<tr>
<td>minFree</td>
<td><code>2000</code></td>
</tr>
<tr>
<td>site</td>
<td><code>bl=10204 lr=255 rec=vb cy pri=1 sec=5</code></td>
</tr>
<tr>
<td>maxRetries</td>
<td><code>72</code></td>
</tr>
<tr>
<td>retInterval</td>
<td><code>60</code></td>
</tr>
</tbody>
</table>

**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`.

---

Chapter 9. Installing the UNIX and Linux Perf.comp.
Considering which Linux subcomponents to install

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 dr1gather 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:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>perfInterval</td>
<td>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.</td>
</tr>
<tr>
<td>perfCount</td>
<td>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.dyyymmd.d, as follows:</td>
</tr>
<tr>
<td></td>
<td>• The minimum, maximum and average values during the perfCount interval</td>
</tr>
<tr>
<td></td>
<td>• The number of users and processes</td>
</tr>
<tr>
<td></td>
<td>• The usage of paging space</td>
</tr>
<tr>
<td></td>
<td>The default interval between samples is 10 minutes.</td>
</tr>
<tr>
<td>diskInterval</td>
<td>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.dyyymmd.d. The default interval between samples is 60 minutes.</td>
</tr>
</tbody>
</table>

Parameter file values for performance component

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>perfInterval</td>
<td>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.</td>
</tr>
<tr>
<td>perfCount</td>
<td>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.dyyymmd.d, as follows:</td>
</tr>
<tr>
<td></td>
<td>• The minimum, maximum and average values during the perfCount interval</td>
</tr>
<tr>
<td></td>
<td>• The number of users and processes</td>
</tr>
<tr>
<td></td>
<td>• The usage of paging space</td>
</tr>
<tr>
<td></td>
<td>The default interval between samples is 10 minutes.</td>
</tr>
<tr>
<td>diskInterval</td>
<td>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.dyyymmd.d. The default interval between samples is 60 minutes.</td>
</tr>
</tbody>
</table>
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.dymmmdd
  - node_name.disk.dymmmdd

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.
Commands to use if Linux Performance component installed

**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.
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.
Planning the implementation process

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 chapter 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 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>
<thead>
<tr>
<th>Installation task</th>
<th>Disk space (approximate)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loading code in the file system, e.g., c:\tdszwin\bin:</td>
<td>1MB</td>
</tr>
<tr>
<td>Creating configuration file in, for example, c:\tdszwin\etc:</td>
<td>1MB</td>
</tr>
<tr>
<td>Creating directory for data files in, for example, c:\tdszwin\var:</td>
<td>10MB (see table note)</td>
</tr>
</tbody>
</table>

**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.
Installing the Windows component on your Windows nodes

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 DRLxxx.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.sdr1ws(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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NodeName</td>
<td>The unique identifier for this node (up to 16 characters).</td>
</tr>
<tr>
<td>XmitServer</td>
<td>Host name (TCP nodename or TCP internet address) of the z/OS host to receive the Windows component data files.</td>
</tr>
<tr>
<td>XmitUser</td>
<td>User name to use for transmitting to host.</td>
</tr>
<tr>
<td>XmitPassword</td>
<td>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.</td>
</tr>
<tr>
<td>DrlPerfdXlat</td>
<td>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):</td>
</tr>
</tbody>
</table>

{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[5,7].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.

   ![Figure 40. Tivoli Decision Support for z/OS Administration window](image)

   The Components window is displayed, (as shown in Figure 41 on page 124).
Installing the Windows component on your Windows nodes

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).

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.
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 DRLXXX.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.

```plaintext
//JOBCLASS  
///  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  
//DRLIN DD DISP=SHR,DSN=db2loadlibrary  
//DRLLOG 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:
Testing the installation

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.

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 chapter topic 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

The processing steps shown in Figure 45 are:

1. Agents gather data into log files
Distributed Systems Performance component general data flow

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).
Distributed Systems Performance component general data flow

- 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>
<thead>
<tr>
<th>Performance component subcomponent</th>
<th>Log file (and record sub-types, when used)</th>
<th>Tivoli Decision Support for z/OS record definition and description (see table note 2)</th>
<th>Tivoli Decision Support for z/OS log definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNIX Accounting</td>
<td>PACCT (command accounting)</td>
<td>XACCT_COMMAND (command accounting data)</td>
<td>UNIX</td>
</tr>
<tr>
<td>UNIX Accounting</td>
<td>WTMP (connect time accounting)</td>
<td>XACCT_CONNECT (user accounting data)</td>
<td>UNIX</td>
</tr>
<tr>
<td>UNIX Accounting</td>
<td>DTMP (disk accounting)</td>
<td>XACCT_DISK (disk usage data)</td>
<td>UNIX</td>
</tr>
<tr>
<td>UNIX Accounting</td>
<td>QACCT (print accounting)</td>
<td>XACCT_PRINT (print accounting data)</td>
<td>UNIX</td>
</tr>
<tr>
<td>UNIX Configuration</td>
<td>CONF (A,D,S) (configuration hardware)</td>
<td>XCONFIG_HW (configuration of hardware data)</td>
<td>UNIX</td>
</tr>
<tr>
<td>UNIX Configuration</td>
<td>CONF (H,I,L) (configuration software)</td>
<td>XCONFIG_SW (configuration of software data)</td>
<td>UNIX</td>
</tr>
<tr>
<td>UNIX Error</td>
<td>ERRPT (error)</td>
<td>XERROR (error log data)</td>
<td>UNIX</td>
</tr>
<tr>
<td>UNIX Performance</td>
<td>PERF (VM) (processor performance)</td>
<td>XPERF_CPU (processor usage data)</td>
<td>UNIX</td>
</tr>
<tr>
<td>UNIX Performance</td>
<td>DISK (disk performance)</td>
<td>XPERF_DISK (disk space usage data)</td>
<td>UNIX</td>
</tr>
<tr>
<td>UNIX Performance</td>
<td>PERF (IO) (disk I/O performance)</td>
<td>XPERF_DISKIO (disk I/O data)</td>
<td>UNIX</td>
</tr>
<tr>
<td>UNIX Performance</td>
<td>PERF (PS) (paging performance)</td>
<td>XPERF_PAGING (paging space usage data)</td>
<td>UNIX</td>
</tr>
<tr>
<td>Windows Performance</td>
<td>NA</td>
<td>WINDOWS_PERF (Windows Performance)</td>
<td>WINDOWS_LOG</td>
</tr>
</tbody>
</table>
### Table 21. Record and log definitions used by the Distributed Systems Performance component (continued)

<table>
<thead>
<tr>
<th>Performance component subcomponent</th>
<th>Log file (and record sub-types, when used)</th>
<th>Tivoli Decision Support for z/OS record definition and description (see table note 2)</th>
<th>Tivoli Decision Support for z/OS log definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Device</td>
<td>NA</td>
<td>WINDOWS_DISK (Windows Disk)</td>
<td>WINDOWS_LOG</td>
</tr>
</tbody>
</table>

**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:

\textit{X\textsubscript{subcomponent} \textit{content} \textit{suf\textsubscript{fix}} or WIN\_PERF \textit{subcomponent} \textit{suf\textsubscript{fix}}}  

where:
• \textit{subcomponent} identifies the subcomponent to which the table belongs (for example, \textit{ACCT} for the accounting subcomponent).
• \textit{content} is a description (for example, \textit{XACCT\_COMMAND} for the accounting subcomponent command statistics).
• \textit{suf\textsubscript{fix}} indicates the summarization level of the data in the table (for example, \textit{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:

\textit{\_H} The table holds data summarized by \textbf{hour} (hourly data).
\textit{\_D} The table holds data summarized by \textbf{day} (daily data).
\textit{\_M} The table holds data summarized by \textbf{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 \textit{views} use the format consisting of the table name from which the view is taken, followed by a \textit{V}. For example, the view \textit{XACCT\_DISK\_MV} is taken from the table \textit{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.”
- “XACCT_DISK_D, _M” on page 135.
- “XACCT_PRINT_D, _M” on page 136.
- “XACCT_CONNECT_D, _M” on page 136.
- “XACCT_DISK_MV” on page 137.

**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

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>K</td>
<td>DATE</td>
</tr>
<tr>
<td>NODE_NAME</td>
<td>K</td>
<td>CHAR(16) Node name. From NODE_NAME.</td>
</tr>
<tr>
<td>PERIOD_NAME</td>
<td>K</td>
<td>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.</td>
</tr>
<tr>
<td>USER_GROUP</td>
<td>K</td>
<td>CHAR(8) Group name. From AC_GID.</td>
</tr>
</tbody>
</table>
### Tables in the UNIX accounting subcomponent

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NODE_GROUP</td>
<td>K</td>
<td>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 '?'.</td>
</tr>
<tr>
<td>USER_NAME</td>
<td>K</td>
<td>User name. From AC_UID.</td>
</tr>
<tr>
<td>TTY</td>
<td>K</td>
<td>Control typewriter. From AC_TTY.</td>
</tr>
<tr>
<td>COMMAND</td>
<td>K</td>
<td>Command name. From AC_COMM.</td>
</tr>
<tr>
<td>COMMANDS</td>
<td>INTEGER</td>
<td>Total number of commands. This is the count of records.</td>
</tr>
<tr>
<td>ELAPSED_SEC</td>
<td>FLOAT</td>
<td>Total elapsed time, in seconds. Calculated as the sum of AC_ETIME/64.</td>
</tr>
<tr>
<td>IO_CHARS</td>
<td>INTEGER</td>
<td>Total characters transferred. This is the sum of AC_IO.</td>
</tr>
<tr>
<td>MEMORY_BYTES</td>
<td>FLOAT</td>
<td>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.</td>
</tr>
<tr>
<td>MEMORY_KB_SEC</td>
<td>FLOAT</td>
<td>Total memory usage, in kilobyte-seconds. Calculated as the sum of (AC_MEM/1024)*(AC_ETIME/64).</td>
</tr>
<tr>
<td>RW_BLOCKS</td>
<td>INTEGER</td>
<td>Total blocks read/written. This is the sum of AC_RW.</td>
</tr>
<tr>
<td>SYSTEM_SEC</td>
<td>FLOAT</td>
<td>Total system time, in seconds. Calculated as the sum of AC_STIME/64.</td>
</tr>
<tr>
<td>USER_SEC</td>
<td>FLOAT</td>
<td>Total user time, in seconds. Calculated as the sum of AC_UTIME/64.</td>
</tr>
</tbody>
</table>

### 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

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>DATE</td>
<td>Date when the records were written. For XACCT_DISK_M, this is the first day of the month. From DTE.</td>
</tr>
<tr>
<td>NODE_NAME</td>
<td>CHAR(16)</td>
<td>Node name. From NODE_NAME.</td>
</tr>
<tr>
<td>USER_NAME</td>
<td>CHAR(8)</td>
<td>User name. From USER_NAME.</td>
</tr>
<tr>
<td>NODE_GROUP</td>
<td>CHAR(16)</td>
<td>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 '?'.</td>
</tr>
<tr>
<td>DISK_BLOCKS_AVG</td>
<td>FLOAT</td>
<td>Average number of disk blocks. This is the average of DISK_BLOCKS over RECORDS_COLLECTED.</td>
</tr>
<tr>
<td>DISK_BLOCKS_SUM</td>
<td>FLOAT</td>
<td>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.</td>
</tr>
<tr>
<td>RECORDS_COLLECTED</td>
<td>INTEGER</td>
<td>Number of records. This is the count of records.</td>
</tr>
</tbody>
</table>
Tables in the UNIX accounting subcomponent

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

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>DATE</td>
<td>Date when the records were written. For XACCT_PRINT_M, this is the first day of the month. From DTE.</td>
</tr>
<tr>
<td>NODE_NAME</td>
<td>CHAR(16)</td>
<td>Node name. From NODE_NAME.</td>
</tr>
<tr>
<td>PERIOD_NAME</td>
<td>CHAR(8)</td>
<td>Name of the period. This is derived using fields NODE_NAME, DTE and TME from the record as parameters in the PERIOD function.</td>
</tr>
<tr>
<td>USER_NAME</td>
<td>CHAR(8)</td>
<td>User name. From the first word of PR_USER.</td>
</tr>
<tr>
<td>PRINT_QUEUE</td>
<td>CHAR(8)</td>
<td>Print queue name. From PRINT_QUEUE.</td>
</tr>
<tr>
<td>PRINT_TYPE</td>
<td>CHAR(8)</td>
<td>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.</td>
</tr>
<tr>
<td>NODE_GROUP</td>
<td>CHAR(16)</td>
<td>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 '?'.</td>
</tr>
<tr>
<td>REQUEST_NODE</td>
<td>CHAR(32)</td>
<td>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.</td>
</tr>
<tr>
<td>PAGES</td>
<td>INTEGER</td>
<td>Number of pages. This is the sum of PAGES.</td>
</tr>
<tr>
<td>PRINTOUTS</td>
<td>INTEGER</td>
<td>Number of printouts. This is the count of records.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>DATE</td>
<td>Date when the records were written. For XACCT_CONNECT_M, this is the first day of the month.</td>
</tr>
<tr>
<td>NODE_NAME</td>
<td>CHAR(16)</td>
<td>Node name. From NODE_NAME.</td>
</tr>
<tr>
<td>PERIOD_NAME</td>
<td>CHAR(8)</td>
<td>Name of the period. This is derived using fields NODE_NAME, DTE and TME from the record as parameters in the PERIOD function.</td>
</tr>
<tr>
<td>USER_NAME</td>
<td>CHAR(8)</td>
<td>User login name. From UT_USER.</td>
</tr>
<tr>
<td>CONNECT_TYPE</td>
<td>CHAR(2)</td>
<td>Connect type code. From UT_TYPE.</td>
</tr>
<tr>
<td>CONNECTS</td>
<td>INTEGER</td>
<td>Total number of connections. This is the count of records.</td>
</tr>
</tbody>
</table>
Tables in the UNIX accounting subcomponent

**XACCT_DISK_MV**

This view provides monthly UNIX disk accounting statistics for the 'root' user. It is based on the XACCT_DISK_M table.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>DATE</td>
<td>Date when the record was written. From DTE. This is the date of the first day of the month.</td>
</tr>
<tr>
<td>NODE_NAME</td>
<td>CHAR(16)</td>
<td>Node name. From NODE_NAME.</td>
</tr>
<tr>
<td>RECORDS_COLLECTED</td>
<td>INTEGER</td>
<td>Number of records. This is the count of records.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>K</td>
<td>DATE Date when the record was written. From DTE.</td>
</tr>
<tr>
<td>TIME</td>
<td>K</td>
<td>TIME Time when the record was written. From TME.</td>
</tr>
<tr>
<td>NODE_NAME</td>
<td>K</td>
<td>CHAR(16) Node name. From NODE_NAME.</td>
</tr>
<tr>
<td>DEVICE_CLASS</td>
<td>K</td>
<td>CHAR(16) Device class. From DEVICE_CLASS.</td>
</tr>
<tr>
<td>DEVICE_SUBCLASS</td>
<td>K</td>
<td>CHAR(16) Device subclass. From DEVICE_SUBC.</td>
</tr>
<tr>
<td>DEVICE_NAME</td>
<td>K</td>
<td>CHAR(16) Device name. From DEVICE_NAME.</td>
</tr>
<tr>
<td>DEVICE_TYPE</td>
<td>K</td>
<td>CHAR(16) Device type. From DEVICE_TYPE.</td>
</tr>
<tr>
<td>STATUS</td>
<td>K</td>
<td>CHAR(2) Device status. From RECORD_SUBTYPE.</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>K</td>
<td>DATE Date when the record was written. From DTE.</td>
</tr>
<tr>
<td>TIME</td>
<td>K</td>
<td>TIME Time when the record was written. From TME.</td>
</tr>
<tr>
<td>NODE_NAME</td>
<td>K</td>
<td>CHAR(16) Node name. From NODE_NAME.</td>
</tr>
<tr>
<td>SOFTWARE_OBJECT</td>
<td>K</td>
<td>CHAR(32) Software object name. From FILE_NAME.</td>
</tr>
<tr>
<td>PATH</td>
<td>K</td>
<td>CHAR(32) Path name for object. From FILE_NAME.</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td></td>
<td>VARCHAR(80) Product description. From FILE_NAME.</td>
</tr>
</tbody>
</table>
Tables in the configuration subcomponent

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FEATURE</td>
<td>CHAR(4)</td>
<td>Feature ID. From FEATURE_ID.</td>
</tr>
<tr>
<td>INSTALL_DATE</td>
<td>CHAR(8)</td>
<td>Product installation date. From INST_DATE.</td>
</tr>
<tr>
<td>INSTALL_STATUS</td>
<td>CHAR(10)</td>
<td>Product installation status. From STATUS.</td>
</tr>
<tr>
<td>PRODUCT_ID</td>
<td>CHAR(10)</td>
<td>Product ID. From PRODUCT_ID.</td>
</tr>
<tr>
<td>PRODUCT_NAME</td>
<td>CHAR(20)</td>
<td>Product name. From PRODUCT_NAME.</td>
</tr>
<tr>
<td>PRODUCT_STATE</td>
<td>CHAR(12)</td>
<td>Product state. From STATE.</td>
</tr>
<tr>
<td>RELEASE</td>
<td>CHAR(15)</td>
<td>Product release number. From RELEASE.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>K</td>
<td>DATE when the error occurred. From DTE. For XERROR_M, this is the first day of the month.</td>
</tr>
<tr>
<td>NODE_NAME</td>
<td>K CHAR(16)</td>
<td>Node name. From NODE_NAME.</td>
</tr>
<tr>
<td>PERIOD_NAME</td>
<td>K CHAR(8)</td>
<td>Name of the period. This is derived using fields NODE_NAME, DTE and TME from the record as parameters in the PERIOD function.</td>
</tr>
<tr>
<td>ERROR_ID</td>
<td>K CHAR(8)</td>
<td>Error identification. From ERROR_ID.</td>
</tr>
<tr>
<td>ERROR_TYPE</td>
<td>K CHAR(1)</td>
<td>Error type. From ERROR_TYPE.</td>
</tr>
<tr>
<td>ERROR_CLASS</td>
<td>K CHAR(1)</td>
<td>Error class. From ERROR_CLASS.</td>
</tr>
<tr>
<td>RESOURCE_NAME</td>
<td>K CHAR(14)</td>
<td>Resource name. From RESOURCE_NAME.</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>VARCHAR(40)</td>
<td>Error description. From DESCRIPTION.</td>
</tr>
<tr>
<td>ERRORS</td>
<td>INTEGER</td>
<td>Number of errors. This is the count of records.</td>
</tr>
</tbody>
</table>

Tables in the UNIX performance subcomponent

This section describes the performance subcomponent tables:
- “XPERF_CPU_H, _D, _M” on page 139
- “XPERF_DISK_D, _M” on page 140
- “XPERF_DISKIO_H, _D, _M” on page 140
- “XPERF_PAGING_H, _D, _M” on page 141
- “XPERF_PS_INFO” on page 142
- “XPERF_VM_INFO” on page 142
- “WIN_PERF_DD_H, _D, _M” on page 143
- “WIN_PERF_PU_H, _D, _M” on page 143
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**

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>K</td>
<td>DATE when the records were written. For XPERF_CPU_M, this is the first day of the month. From DTE.</td>
</tr>
<tr>
<td>TIME</td>
<td>K</td>
<td>TIME (rounded down to the nearest hour) when the record was written. From TME. This field is present only in XPERF_CPU_H.</td>
</tr>
<tr>
<td>NODE_NAME</td>
<td>K</td>
<td>CHAR(16) Node name. From NODE_NAME.</td>
</tr>
<tr>
<td>PERIOD_NAME</td>
<td>K</td>
<td>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.</td>
</tr>
<tr>
<td>CPU_USAGE_PCT_AVG</td>
<td>FLOAT</td>
<td>Average CPU usage, in percent. This is the average of CPU_AVG.</td>
</tr>
<tr>
<td>CPU_USAGE_PCT_MAX</td>
<td>INTEGER</td>
<td>Maximum CPU usage, in percent. This is the maximum of CPU_MAX.</td>
</tr>
<tr>
<td>CPU_USAGE_PCT_MIN</td>
<td>INTEGER</td>
<td>Minimum CPU usage, in percent. This is the minimum of CPU_MIN.</td>
</tr>
<tr>
<td>MEASURED_SEC</td>
<td>INTEGER</td>
<td>Total measured time, in seconds. This is the sum of INTERVAL*60.</td>
</tr>
<tr>
<td>MEM_FREE_PAGES_AVG</td>
<td>FLOAT</td>
<td>Average number of free pages. This is the average of MEMORY_AVG.</td>
</tr>
<tr>
<td>MEM_FREE_PAGES_MAX</td>
<td>INTEGER</td>
<td>Maximum number of free pages. This is the maximum of MEMORY_MAX.</td>
</tr>
<tr>
<td>MEM_FREE_PAGES_MIN</td>
<td>INTEGER</td>
<td>Minimum number of free pages. This is the minimum of MEMORY_MIN.</td>
</tr>
<tr>
<td>PAGING_RATE_AVG</td>
<td>FLOAT</td>
<td>Average number of pages paged IN/OUT per second. This is the average of PAGING_AVG.</td>
</tr>
<tr>
<td>PAGING_RATE_MAX</td>
<td>INTEGER</td>
<td>Maximum number of pages paged IN/OUT per second. This is the maximum of PAGING_MAX.</td>
</tr>
<tr>
<td>PAGING_RATE_MIN</td>
<td>INTEGER</td>
<td>Minimum number of pages paged IN/OUT per second. This is the minimum of PAGING_MIN.</td>
</tr>
<tr>
<td>PROCESSES_AVG</td>
<td>FLOAT</td>
<td>Average number of processes. This is the average of PROCESSES.</td>
</tr>
<tr>
<td>PROCESSES_MAX</td>
<td>INTEGER</td>
<td>Maximum number of processes. This is the maximum of PROCESSES.</td>
</tr>
<tr>
<td>PROCESSES_MIN</td>
<td>INTEGER</td>
<td>Minimum number of processes. This is the minimum of PROCESSES.</td>
</tr>
<tr>
<td>RECORDS_COLLECTED</td>
<td>INTEGER</td>
<td>Total number of records. This is the count of records.</td>
</tr>
<tr>
<td>USERS_AVG</td>
<td>FLOAT</td>
<td>Average number of users. This is the average of USERS.</td>
</tr>
<tr>
<td>USERS_MAX</td>
<td>INTEGER</td>
<td>Maximum number of users. This is the maximum of USERS.</td>
</tr>
<tr>
<td>USERS_MIN</td>
<td>INTEGER</td>
<td>Minimum number of users. This is the minimum of USERS.</td>
</tr>
</tbody>
</table>
**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

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>DATE</td>
<td>Date when the records were written. For XPERF_DISK_M, this is the first day</td>
</tr>
<tr>
<td></td>
<td></td>
<td>of the month. From DTE.</td>
</tr>
<tr>
<td>NODE_NAME</td>
<td>CHAR(16)</td>
<td>Node name. From NODE_NAME.</td>
</tr>
<tr>
<td>PERIOD_NAME</td>
<td>CHAR(8)</td>
<td>Name of the period. This is derived using fields NODE_NAME, DTE and TME from</td>
</tr>
<tr>
<td></td>
<td></td>
<td>the record as parameters in the PERIOD function.</td>
</tr>
<tr>
<td>SPACE_TYPE</td>
<td>CHAR(2)</td>
<td>Space type, FS or VG. From RECORD_SUBTYPE.</td>
</tr>
<tr>
<td>VOLUME</td>
<td>CHAR(16)</td>
<td>Volume. From FILE_SYSTEM for space type FS and from VOLUME_GROUP for space</td>
</tr>
<tr>
<td></td>
<td></td>
<td>type VG.</td>
</tr>
<tr>
<td>FILE_SYSTEM</td>
<td>CHAR(16)</td>
<td>File system. From FILE_SYSTEM for space type FS and set to blanks for space</td>
</tr>
<tr>
<td></td>
<td></td>
<td>type VG.</td>
</tr>
<tr>
<td>RECORDS_COLLECTED</td>
<td>INTEGER</td>
<td>Number of samples. This is the count of records.</td>
</tr>
<tr>
<td>SPACE_FREE_MB</td>
<td>FLOAT</td>
<td>Free space, in MB, accumulated for all samples. This is the sum of</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FREE_SPACE_MB for space type VG and calculated as the sum of FREE_SPACE_KB/</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1024.0 for space type FS. This value should be divided by RECORDS_COLLECTED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>to get the average free space.</td>
</tr>
<tr>
<td>SPACE_SIZE_MB</td>
<td>FLOAT</td>
<td>Total size of space, in MB, accumulated for all samples. This is the sum</td>
</tr>
<tr>
<td></td>
<td></td>
<td>of TOTAL_SPACE_MB for space type VG and calculated as the sum of TOTAL_SPACE_KB/</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1024.0 for space type FS. Divide this value by RECORDS_COLLECTED to get the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>average value.</td>
</tr>
<tr>
<td>SPACE_USED_PCT</td>
<td>FLOAT</td>
<td>Used space, in percent, accumulated for all samples. Calculated as the sum</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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.</td>
</tr>
</tbody>
</table>

**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
### Tables in the performance subcomponent

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>DATE</td>
<td>Date when the records were written. For XPERF_DISKIO_M, this is the first day of the month. From DTE.</td>
</tr>
<tr>
<td>TIME</td>
<td>TIME</td>
<td>Time (rounded down to the nearest hour) when the record was written. From TME. This field is present only for XPERF_DISKIO_H.</td>
</tr>
<tr>
<td>NODE_NAME</td>
<td>CHAR(16)</td>
<td>Node name. From NODE_NAME.</td>
</tr>
<tr>
<td>PERIOD_NAME</td>
<td>CHAR(8)</td>
<td>Name of the period. This is derived using fields NODE_NAME, DTE and TME from the record as parameters in the PERIOD function.</td>
</tr>
<tr>
<td>DISK</td>
<td>CHAR(10)</td>
<td>Physical disk name. From DISK.</td>
</tr>
<tr>
<td>DISK_BUSY_PCT_AVG</td>
<td>FLOAT</td>
<td>Average disk busy, in percent. This is the average of BUSY.</td>
</tr>
<tr>
<td>DISK_BUSY_PCT_MAX</td>
<td>INTEGER</td>
<td>Maximum disk busy, in percent. This is the maximum of BUSY.</td>
</tr>
<tr>
<td>MEASURED_SEC</td>
<td>INTEGER</td>
<td>Total measurement time, in seconds. Calculated as the sum of INTERVAL*60.</td>
</tr>
<tr>
<td>READ_KB_TOTAL</td>
<td>INTEGER</td>
<td>Amount of data read, in kilobytes. This is the sum of TOTREAD.</td>
</tr>
<tr>
<td>READ_MAXRATE</td>
<td>FLOAT</td>
<td>Maximum amount of data read, in kilobytes per second. Calculated as the maximum of TOTREAD/(INTERVAL*60.0).</td>
</tr>
<tr>
<td>RECORDS_COLLECTED</td>
<td>INTEGER</td>
<td>Number of records. This is the count of records.</td>
</tr>
<tr>
<td>WRITE_KB_TOTAL</td>
<td>INTEGER</td>
<td>Amount of data written, in kilobytes. This is the sum of TOTWRITE.</td>
</tr>
<tr>
<td>WRITE_MAXRATE</td>
<td>FLOAT</td>
<td>Maximum amount of data written, in kilobytes per second. Calculated as the maximum of TOTWRITE/(INTERVAL*60.0).</td>
</tr>
<tr>
<td>RW_KB_TOTAL</td>
<td>FLOAT</td>
<td>Amount of data written, in kilobytes. Calculated as the sum of TOTRW.</td>
</tr>
<tr>
<td>RW_MAXRATE</td>
<td>INTEGER</td>
<td>Maximum amount of data written, in kilobytes per second. Calculated as the sum of TOTRW/(INTERVAL*60.0).</td>
</tr>
</tbody>
</table>

**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

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>K</td>
<td>Date when the records were written. For XPERF_PAGING_M, this is the first day of the month. From DTE.</td>
</tr>
<tr>
<td>TIME</td>
<td>K</td>
<td>Time (rounded down to the nearest hour) when the record was written. From TME. This field is present only for XPERF_PAGING_H.</td>
</tr>
<tr>
<td>NODE_NAME</td>
<td>K CHAR(16)</td>
<td>Node name. From NODE_NAME.</td>
</tr>
</tbody>
</table>
### Tables in the performance subcomponent

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PERIOD_NAME</td>
<td>K</td>
<td>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.</td>
</tr>
<tr>
<td>DISK</td>
<td>K</td>
<td>CHAR(10) Physical disk name. From DISK.</td>
</tr>
<tr>
<td>PAGE_SPACE</td>
<td>K</td>
<td>CHAR(10) Page space name. From PAGE_SPACE.</td>
</tr>
<tr>
<td>PS_SIZE_AVG_KB</td>
<td>FLOAT</td>
<td>Average page space size, in KB. This is the average of PSSIZE.</td>
</tr>
<tr>
<td>PS_USED_AVG_KB</td>
<td>FLOAT</td>
<td>Average page space used, in KB. This is the average of PSUSE.</td>
</tr>
<tr>
<td>RECORDS_COLLECTED</td>
<td>INTEGER</td>
<td>Number of records. This is the count of records.</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>DATE</td>
<td>The date when the record was written. From DTE.</td>
</tr>
<tr>
<td>NODE_NAME</td>
<td>CHAR(16)</td>
<td>The node identification.</td>
</tr>
<tr>
<td>PERIOD_NAME</td>
<td>CHAR(8)</td>
<td>The name of the period.</td>
</tr>
<tr>
<td>PHY_VOL</td>
<td>CHAR(20)</td>
<td>The name of the physical volume.</td>
</tr>
<tr>
<td>VOL_SIZE</td>
<td>INTEGER</td>
<td>The size of the volume.</td>
</tr>
<tr>
<td>BLOCKS</td>
<td>INTEGER</td>
<td>The number of blocks that are used in the volume.</td>
</tr>
<tr>
<td>SYSTEM</td>
<td>CHAR(5)</td>
<td>UNIX or LINUX system.</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>DATE</td>
<td>The date when the record was written. From DTE.</td>
</tr>
<tr>
<td>NODE_NAME</td>
<td>CHAR(16)</td>
<td>The node identification.</td>
</tr>
<tr>
<td>PERIOD_NAME</td>
<td>CHAR(8)</td>
<td>The name of the period.</td>
</tr>
<tr>
<td>USER</td>
<td>INTEGER</td>
<td>The number of users.</td>
</tr>
<tr>
<td>PROC</td>
<td>INTEGER</td>
<td>The number of processes.</td>
</tr>
<tr>
<td>DUR_INT</td>
<td>INTEGER</td>
<td>The duration of the interval.</td>
</tr>
<tr>
<td>MEM_MIN_U</td>
<td>INTEGER</td>
<td>The minimum memory user page.</td>
</tr>
<tr>
<td>AVG_MEM_U</td>
<td>FLOAT</td>
<td>The average memory user page.</td>
</tr>
<tr>
<td>MEM_MAX_U</td>
<td>INTEGER</td>
<td>The maximum memory user page.</td>
</tr>
<tr>
<td>MEM_MIN_S</td>
<td>INTEGER</td>
<td>The minimum memory page swap.</td>
</tr>
<tr>
<td>AVG_MEM_S</td>
<td>FLOAT</td>
<td>The average memory page swap.</td>
</tr>
<tr>
<td>MEM_MAX_S</td>
<td>INTEGER</td>
<td>The maximum memory page swap.</td>
</tr>
<tr>
<td>CPU_MIN</td>
<td>INTEGER</td>
<td>The minimum CPU value.</td>
</tr>
<tr>
<td>AVG_CPU</td>
<td>FLOAT</td>
<td>The average CPU value.</td>
</tr>
</tbody>
</table>
Tables in the performance subcomponent

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU_MAX</td>
<td>INTEGER</td>
<td>The maximum CPU value.</td>
</tr>
<tr>
<td>SYSTEM</td>
<td>CHAR(5)</td>
<td>UNIX or LINUX system.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Column</th>
<th>Key</th>
<th>Datatype</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>K</td>
<td>DATE</td>
<td>The date of the record. For the _M table, dates are stored as the first day of the month.</td>
</tr>
<tr>
<td>TIME</td>
<td>K</td>
<td>TIME</td>
<td>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.</td>
</tr>
<tr>
<td>NODE_NAME</td>
<td>K</td>
<td>CHAR(16)</td>
<td>The node name of the machine.</td>
</tr>
<tr>
<td>DEVICE_NAME</td>
<td>K</td>
<td>CHAR(8)</td>
<td>The device name for the device.</td>
</tr>
<tr>
<td>SAMPLES</td>
<td></td>
<td>INTEGER</td>
<td>Count of the records consolidated into a table row. Used for working out averages.</td>
</tr>
<tr>
<td>TOTAL_DEVICE_SIZE</td>
<td></td>
<td>FLOAT</td>
<td>Device size at the end of the period.</td>
</tr>
<tr>
<td>FREE_SPACE_AVG</td>
<td></td>
<td>FLOAT</td>
<td>Averages the free space on the device, based on SAMPLES.</td>
</tr>
<tr>
<td>FREE_SPACE_MIN</td>
<td></td>
<td>FLOAT</td>
<td>The minimum free space available.</td>
</tr>
<tr>
<td>FREE_SPACE_MAX</td>
<td></td>
<td>FLOAT</td>
<td>The maximum free space available.</td>
</tr>
<tr>
<td>PERCENT_FREE_AVG</td>
<td></td>
<td>FLOAT</td>
<td>Averages the free space percentage on the device, based on SAMPLES.</td>
</tr>
<tr>
<td>PERCENT_FREE_MIN</td>
<td></td>
<td>FLOAT</td>
<td>The minimum free space percentage</td>
</tr>
<tr>
<td>PERCENT_FREE_MAX</td>
<td></td>
<td>FLOAT</td>
<td>The maximum free space percentage.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Column</th>
<th>Key</th>
<th>Datatype</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>K</td>
<td>DATE</td>
<td>The date of the record. For the _M table, dates are forced back to the first day of the month.</td>
</tr>
<tr>
<td>TIME</td>
<td>K</td>
<td>TIME</td>
<td>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.</td>
</tr>
</tbody>
</table>
Tables in the Windows component

<table>
<thead>
<tr>
<th>Column</th>
<th>Key</th>
<th>Datatype</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NODE_NAME</td>
<td>K</td>
<td>CHAR(16)</td>
<td>The node name of the machine.</td>
</tr>
<tr>
<td>SAMPLES</td>
<td></td>
<td>INTEGER</td>
<td>Count of the records consolidated into a table row. Used for working out actuals.</td>
</tr>
<tr>
<td>CPU_COUNT</td>
<td></td>
<td>SMALLINT</td>
<td>Number of CPUs and/or cores.</td>
</tr>
<tr>
<td>CPU_USAGE_AVG</td>
<td></td>
<td>FLOAT</td>
<td>Averages the average CPU usage. This figure is averaged based on SAMPLES.</td>
</tr>
<tr>
<td>CPU_USAGE_MIN</td>
<td></td>
<td>FLOAT</td>
<td>The minimum CPU usage.</td>
</tr>
<tr>
<td>CPU_USAGE_MAX</td>
<td></td>
<td>FLOAT</td>
<td>The maximum CPU usage.</td>
</tr>
<tr>
<td>MEMORY_SIZE</td>
<td></td>
<td>FLOAT</td>
<td>The total memory size at the end of the period.</td>
</tr>
<tr>
<td>MEMORY_FREE_AVG</td>
<td></td>
<td>FLOAT</td>
<td>Averages the average free memory. This figure is averaged based on SAMPLES.</td>
</tr>
<tr>
<td>MEMORY_FREE_MIN</td>
<td></td>
<td>FLOAT</td>
<td>The minimum free memory.</td>
</tr>
<tr>
<td>MEMORY_FREE_MAX</td>
<td></td>
<td>FLOAT</td>
<td>The maximum free memory.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONNECT_TYPE</td>
<td>K</td>
<td>Connect type code.</td>
</tr>
<tr>
<td>CONNECT_DESC</td>
<td>CHAR(16)</td>
<td>Connect type description.</td>
</tr>
</tbody>
</table>

Example of table contents

<table>
<thead>
<tr>
<th>PROCESS_CODE</th>
<th>PROCESS_TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>---------------</td>
<td>--------------</td>
</tr>
<tr>
<td>00</td>
<td>EMPTY</td>
</tr>
<tr>
<td>01</td>
<td>RUN_LVL</td>
</tr>
<tr>
<td>02</td>
<td>BOOT_TIME</td>
</tr>
<tr>
<td>03</td>
<td>OLD_TIME</td>
</tr>
<tr>
<td>04</td>
<td>NEW_TIME</td>
</tr>
<tr>
<td>05</td>
<td>INIT_PROCESS</td>
</tr>
<tr>
<td>06</td>
<td>LOGIN_PROCESS</td>
</tr>
<tr>
<td>07</td>
<td>USER_PROCESS</td>
</tr>
<tr>
<td>08</td>
<td>DEAD_PROCESS</td>
</tr>
<tr>
<td>09</td>
<td>ACCOUNTING</td>
</tr>
</tbody>
</table>
X_NODE_NAME

This lookup table contains node names and their associated accounting groups, period plans and TCP/IP address.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NODE_NAME</td>
<td>K</td>
<td>Node name. This is an alias name for the node. This may contain global search characters.</td>
</tr>
<tr>
<td>ACCOUNTING_GROUP</td>
<td>CHAR(16)</td>
<td>Accounting group.</td>
</tr>
<tr>
<td>INTERNET_ADDRESS</td>
<td>CHAR(15)</td>
<td>Internet address. This may contain global search characters.</td>
</tr>
<tr>
<td>PERIOD_PLAN</td>
<td>CHAR(8)</td>
<td>Period plan. This value can be used as a parameter when using the PERIOD function looking up PERIOD_NAME.</td>
</tr>
</tbody>
</table>

Example of table contents

<table>
<thead>
<tr>
<th>NODE_NAME</th>
<th>ACCOUNTING_GROUP</th>
<th>PERIOD_PLAN</th>
<th>INTERNET_ADDRESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>node1</td>
<td>GROUP1</td>
<td>BASE</td>
<td>9.52.50.7</td>
</tr>
<tr>
<td>node2</td>
<td>GROUP2</td>
<td>PLAN1</td>
<td>9.52.50.8</td>
</tr>
<tr>
<td>node3</td>
<td>GROUP2</td>
<td>PLAN2</td>
<td>9.52.50.9</td>
</tr>
<tr>
<td>node5</td>
<td>GROUP3</td>
<td>PLAN3</td>
<td>9.52.50.11</td>
</tr>
<tr>
<td>%</td>
<td>DEFAULT</td>
<td>DEFAULT</td>
<td>9.99.99.99</td>
</tr>
</tbody>
</table>

XACCT_PRINT_TYPE

This lookup table converts print-queues to a print-type. It updates the XACCT_PRINT_D table.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NODE_NAME</td>
<td>K</td>
<td>Node name. This may contain global search characters.</td>
</tr>
<tr>
<td>PRINT_QUEUE</td>
<td>CHAR(8)</td>
<td>Print queue name. This may contain global search characters.</td>
</tr>
<tr>
<td>PRINT_TYPE</td>
<td>CHAR(8)</td>
<td>Print queue type.</td>
</tr>
</tbody>
</table>

Example of table contents

<table>
<thead>
<tr>
<th>NODE_NAME</th>
<th>PRINT_QUEUE</th>
<th>PRINT_TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>node1</td>
<td>prt01</td>
<td>dep01</td>
</tr>
<tr>
<td>node2</td>
<td>prt01</td>
<td>dep01</td>
</tr>
<tr>
<td>node3</td>
<td>prt05</td>
<td>dep02</td>
</tr>
<tr>
<td>%</td>
<td>prt99</td>
<td>dep99</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>DATE</td>
<td>The date when the record was written. From DTE.</td>
</tr>
<tr>
<td>NODE_NAME</td>
<td>CHAR(16)</td>
<td>The node identification.</td>
</tr>
<tr>
<td>PERIOD_NAME</td>
<td>CHAR(8)</td>
<td>The name of the period.</td>
</tr>
<tr>
<td>DEVICE_CLS</td>
<td>CHAR(10)</td>
<td>The device class of the hardware element (Video, HD, CD).</td>
</tr>
<tr>
<td>CPU</td>
<td>CHAR(10)</td>
<td>The CPU identification.</td>
</tr>
<tr>
<td>ADD_ID</td>
<td>CHAR(10)</td>
<td>The address ID of the UNIX device.</td>
</tr>
<tr>
<td>DEVICE_ASG</td>
<td>CHAR(7)</td>
<td>The type of device used in the UNIX system.</td>
</tr>
<tr>
<td>BUS</td>
<td>CHAR(7)</td>
<td>The type of bus used in the UNIX system.</td>
</tr>
</tbody>
</table>

XCONF_SOFTWARE

This table provides software configuration data for the packages used. It maps the DRLTXCNF record.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>DATE</td>
<td>The date when the record was written. From DTE.</td>
</tr>
<tr>
<td>NODE_NAME</td>
<td>CHAR(16)</td>
<td>The node identification.</td>
</tr>
<tr>
<td>PERIOD_NAME</td>
<td>CHAR(8)</td>
<td>The name of the period.</td>
</tr>
<tr>
<td>PACK_NAME</td>
<td>CHAR(50)</td>
<td>The name of the package used on the system.</td>
</tr>
<tr>
<td>PACK_NAME</td>
<td>CHAR(50)</td>
<td>The name of the package used on the system.</td>
</tr>
<tr>
<td>SOFT_VER</td>
<td>CHAR(12)</td>
<td>The software version of the package installed on the system.</td>
</tr>
<tr>
<td>INST_TS</td>
<td>CHAR(12)</td>
<td>The timestamp of when the software was installed.</td>
</tr>
</tbody>
</table>

XDISK_INFOR

This table provides space allocation data for the disks on the file system. It maps the DRLTXDSK record.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>DATE</td>
<td>The date when the record was written. From DTE.</td>
</tr>
<tr>
<td>NODE_NAME</td>
<td>CHAR(16)</td>
<td>The node identification.</td>
</tr>
<tr>
<td>PERIOD_NAME</td>
<td>CHAR(8)</td>
<td>The name of the period.</td>
</tr>
<tr>
<td>FILE_SYS</td>
<td>CHAR(9)</td>
<td>The name of the file system.</td>
</tr>
<tr>
<td>TOT_SIZE</td>
<td>INTEGER</td>
<td>The maximum size of the disk.</td>
</tr>
<tr>
<td>FREE</td>
<td>INTEGER</td>
<td>The free space on the disk.</td>
</tr>
</tbody>
</table>
**XDTMP_INFOR**

This table provides process information related to users. It maps the DRLTXDTM record.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>DATE</td>
<td>The date when the record was written. From DTE.</td>
</tr>
<tr>
<td>NODE_NAME</td>
<td>CHAR(16)</td>
<td>The node identification.</td>
</tr>
<tr>
<td>PERIOD_NAME</td>
<td>CHAR(8)</td>
<td>The name of the period.</td>
</tr>
<tr>
<td>US_ID</td>
<td>CHAR(9)</td>
<td>The user ID.</td>
</tr>
<tr>
<td>US_NAME</td>
<td>CHAR(20)</td>
<td>The user name.</td>
</tr>
<tr>
<td>USED_BLOCK</td>
<td>CHAR(10)</td>
<td>The number of disk blocks that are used.</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>DATE</td>
<td>The date when the record was written. From DTE.</td>
</tr>
<tr>
<td>NODE_NAME</td>
<td>CHAR(16)</td>
<td>The node identification.</td>
</tr>
<tr>
<td>PERIOD_NAME</td>
<td>CHAR(8)</td>
<td>The name of the period.</td>
</tr>
<tr>
<td>PHY_VOL</td>
<td>CHAR(20)</td>
<td>The name of the physical volume.</td>
</tr>
<tr>
<td>VOL_SIZE</td>
<td>INTEGER</td>
<td>The size of the volume.</td>
</tr>
<tr>
<td>BLOCKS</td>
<td>INTEGER</td>
<td>The number of blocks that are used in the volume.</td>
</tr>
<tr>
<td>SYSTEM</td>
<td>CHAR(5)</td>
<td>UNIX or Linux system.</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>DATE</td>
<td>The date when the record was written. From DTE.</td>
</tr>
<tr>
<td>NODE_NAME</td>
<td>CHAR(16)</td>
<td>The node identification.</td>
</tr>
<tr>
<td>PERIOD_NAME</td>
<td>CHAR(8)</td>
<td>The name of the period.</td>
</tr>
<tr>
<td>USER</td>
<td>INTEGER</td>
<td>The number of users.</td>
</tr>
<tr>
<td>PROC</td>
<td>INTEGER</td>
<td>The number of processes.</td>
</tr>
<tr>
<td>DUR_INT</td>
<td>INTEGER</td>
<td>The duration of the interval.</td>
</tr>
<tr>
<td>MEM_MIN_U</td>
<td>INTEGER</td>
<td>The minimum memory user page.</td>
</tr>
<tr>
<td>AVG_MEM_U</td>
<td>FLOAT</td>
<td>The average memory user page.</td>
</tr>
<tr>
<td>MEM_MAX_U</td>
<td>INTEGER</td>
<td>The maximum memory user page.</td>
</tr>
<tr>
<td>MEM_MIN_S</td>
<td>INTEGER</td>
<td>The minimum memory page swap.</td>
</tr>
<tr>
<td>AVG_MEM_S</td>
<td>FLOAT</td>
<td>The average memory page swap.</td>
</tr>
<tr>
<td>MEM_MAX_S</td>
<td>INTEGER</td>
<td>The maximum memory page swap.</td>
</tr>
</tbody>
</table>
### XPERF_VM_INFO

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU_MIN</td>
<td>INTEGER</td>
<td>The minimum CPU value.</td>
</tr>
<tr>
<td>AVG_CPU</td>
<td>FLOAT</td>
<td>The average CPU value.</td>
</tr>
<tr>
<td>CPU_MAX</td>
<td>INTEGER</td>
<td>The maximum CPU value.</td>
</tr>
<tr>
<td>SYSTEM</td>
<td>CHAR(5)</td>
<td>UNIX or Linux system.</td>
</tr>
</tbody>
</table>

### XWTMP_INFOR

This table provides process information related to users. It maps the DRLTXDTM record.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>DATE</td>
<td>The date when the record was written. From DTE.</td>
</tr>
<tr>
<td>NODE_NAME</td>
<td>CHAR(16)</td>
<td>The node identification.</td>
</tr>
<tr>
<td>PERIOD_NAME</td>
<td>CHAR(8)</td>
<td>The name of the period.</td>
</tr>
<tr>
<td>USER_LOG</td>
<td>CHAR(8)</td>
<td>The user login name.</td>
</tr>
<tr>
<td>ENTRY</td>
<td>CHAR(2)</td>
<td>The entry type.</td>
</tr>
<tr>
<td>PROC_ID</td>
<td>CHAR(6)</td>
<td>The process ID.</td>
</tr>
<tr>
<td>NODE</td>
<td>CHAR(16)</td>
<td>The node name.</td>
</tr>
</tbody>
</table>
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 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

---

```
<table>
<thead>
<tr>
<th>User name</th>
<th>Command name</th>
<th>Commands (count)</th>
<th>Elapsed time (hour)</th>
<th>System time (sec)</th>
<th>User time (sec)</th>
<th>Cmd mem (KB)</th>
<th>I/O (KB)</th>
<th>RW blocks (count)</th>
</tr>
</thead>
<tbody>
<tr>
<td>smith</td>
<td>aixterm</td>
<td>9</td>
<td>0.02</td>
<td>1</td>
<td>3</td>
<td>0.23</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>awk</td>
<td>33</td>
<td>0.09</td>
<td>5</td>
<td>5</td>
<td>0.55</td>
<td>14</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>xset</td>
<td>14</td>
<td>0.10</td>
<td>4</td>
<td>10</td>
<td>0.78</td>
<td>18</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>314</td>
<td>2.19</td>
<td>58</td>
<td>55</td>
<td>5.15</td>
<td>92</td>
<td>80</td>
</tr>
<tr>
<td>root</td>
<td>awk</td>
<td>18</td>
<td>0.80</td>
<td>7</td>
<td>5</td>
<td>0.15</td>
<td>13</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>bsh</td>
<td>38</td>
<td>0.21</td>
<td>8</td>
<td>11</td>
<td>0.89</td>
<td>22</td>
<td>68</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>xmservd</td>
<td>12</td>
<td>1.17</td>
<td>21</td>
<td>24</td>
<td>1.19</td>
<td>51</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>418</td>
<td>12.37</td>
<td>170</td>
<td>177</td>
<td>10.23</td>
<td>195</td>
<td>210</td>
</tr>
</tbody>
</table>
```

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.
Accounting reports

**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
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 Users by Command, Daily Overview

![Tivoli Decision Support for z/OS Report: XACCT02](image)

**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 M” on page 134)

Attributes
UNIX, Acct, Accounting, Command, Cmd, User, Daily, Overview

Variables
Date, Node name, Period name

<table>
<thead>
<tr>
<th>Command name</th>
<th>Commands (count)</th>
<th>Elapsed avg (min)</th>
<th>System avg (sec)</th>
<th>User avg (sec)</th>
<th>Cmd mem avg (KB)</th>
<th>IO avg (block)</th>
<th>RW avg (block)</th>
</tr>
</thead>
<tbody>
<tr>
<td>aixterm</td>
<td>7</td>
<td>0.08</td>
<td>0.23</td>
<td>0.33</td>
<td>0.23</td>
<td>17.0</td>
<td>5</td>
</tr>
<tr>
<td>awk</td>
<td>5</td>
<td>0.03</td>
<td>0.04</td>
<td>0.05</td>
<td>0.13</td>
<td>4.1</td>
<td>2</td>
</tr>
<tr>
<td>bind</td>
<td>8</td>
<td>0.02</td>
<td>0.09</td>
<td>0.64</td>
<td>0.44</td>
<td>12.5</td>
<td>3</td>
</tr>
<tr>
<td>bsh</td>
<td>26</td>
<td>0.02</td>
<td>0.10</td>
<td>0.03</td>
<td>0.07</td>
<td>4.4</td>
<td>1</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
</tbody>
</table>

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).
Accounting reports

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

<table>
<thead>
<tr>
<th>User name</th>
<th>Commands (count)</th>
<th>Elapsed time (hour)</th>
<th>System time (sec)</th>
<th>User time (sec)</th>
<th>IO (KB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>smith</td>
<td>514</td>
<td>1.19</td>
<td>58</td>
<td>125</td>
<td>3392</td>
</tr>
<tr>
<td>root</td>
<td>104</td>
<td>0.18</td>
<td>4</td>
<td>2</td>
<td>317</td>
</tr>
</tbody>
</table>

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.
Accounting reports

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
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

---

<table>
<thead>
<tr>
<th>User name</th>
<th>Disk blocks avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>adm</td>
<td>216</td>
</tr>
<tr>
<td>archer</td>
<td>4</td>
</tr>
<tr>
<td>bin</td>
<td>398644</td>
</tr>
<tr>
<td>smith</td>
<td>188</td>
</tr>
<tr>
<td>daemon</td>
<td>4</td>
</tr>
<tr>
<td>guest</td>
<td>4</td>
</tr>
<tr>
<td>johanbe</td>
<td>4</td>
</tr>
<tr>
<td>load1</td>
<td>116</td>
</tr>
<tr>
<td>root</td>
<td>103824</td>
</tr>
<tr>
<td>sys</td>
<td>8</td>
</tr>
<tr>
<td>uucp</td>
<td>972</td>
</tr>
<tr>
<td>xbill</td>
<td>132</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>504116</strong></td>
</tr>
</tbody>
</table>

Figure 50. Example of UNIX Acct Disk Blocks by User, Monthly Overview
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” 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

Figure 51. Example of UNIX Acct Disk Blocks in 1000s, Monthly Trend

<table>
<thead>
<tr>
<th>Node name</th>
<th>avg (1000s)</th>
<th>1999-10-01</th>
<th>1999-11-01</th>
<th>1999-12-01</th>
</tr>
</thead>
<tbody>
<tr>
<td>aix555</td>
<td>504</td>
<td>684</td>
<td>824</td>
<td></td>
</tr>
</tbody>
</table>

Tivoli Decision Support for z/OS: Distributed Systems Performance Feature Guide and Reference
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 136)

Attributes
UNIX, Acct, Accounting, Print, Page, User, Monthly, Overview

Variables
Month, Node name, Period name
### Accounting reports

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 User, Monthly Overview

<table>
<thead>
<tr>
<th>User name</th>
<th>Print queue name</th>
<th>Request node</th>
<th>Printouts</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>smith</td>
<td>prt0</td>
<td>aix555</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>wn5prt</td>
<td></td>
<td>aix555</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>---------</strong></td>
<td><strong>----------</strong></td>
<td><strong>4</strong></td>
<td><strong>4</strong></td>
</tr>
</tbody>
</table>

**Figure 53. Example of UNIX Acct Printed Pages by User, 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
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)
Configuration reports

Attributes
AIX, Configuration, HW, Hardware, Overview

Variables
From date, To date, Node name

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

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 HW for Device Class, Overview

Date: '2003-11-13' to '2003-12-13'

Device class: 'adapter'

<table>
<thead>
<tr>
<th>Node name</th>
<th>Devices</th>
<th>Devices</th>
</tr>
</thead>
<tbody>
<tr>
<td>aix555</td>
<td>12</td>
<td>15</td>
</tr>
</tbody>
</table>

Tivoli Decision Support for z/OS Report: XCONFIG02

**Figure 56. Example of UNIX Configuration of HW for Device Class, Overview**

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
UNIX Configuration of SW for Object, Overview

Date: '2003-01-01' to '2003-07-01'
Node name: 'aix555'

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:

<table>
<thead>
<tr>
<th>Feature code</th>
<th>Path name</th>
<th>Software object</th>
<th>Install date</th>
<th>Description</th>
<th>Date</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>/usr/lib/objrepo</td>
<td>xmconsole.data</td>
<td>XMconsole configuration files</td>
<td>09/13/03</td>
<td></td>
<td>2003-01-17</td>
<td>09.51.31</td>
</tr>
<tr>
<td>/etc/objrepo</td>
<td>xlcmp.obj</td>
<td>AIX XL C Compiler/6000</td>
<td>09/13/03</td>
<td></td>
<td>2003-01-17</td>
<td>09.51.31</td>
</tr>
<tr>
<td>/usr/lib/objrepo</td>
<td>xlcmp.obj</td>
<td>AIX XL C Compiler/6000</td>
<td>09/13/03</td>
<td></td>
<td>2003-01-17</td>
<td>09.51.31</td>
</tr>
<tr>
<td>/etc/objrepo</td>
<td>netmgrEm.Ul.msg</td>
<td>Network Management I Messages</td>
<td>09/13/03</td>
<td></td>
<td>2003-01-17</td>
<td>09.51.31</td>
</tr>
<tr>
<td>/etc/objrepo</td>
<td>netmgr.am.obj</td>
<td>Alert Manager</td>
<td>09/13/03</td>
<td></td>
<td>2003-01-17</td>
<td>09.51.31</td>
</tr>
<tr>
<td>/etc/objrepo</td>
<td>netmgr.api.obj</td>
<td>SNMP Appl. Programming Interf</td>
<td>09/13/03</td>
<td></td>
<td>2003-01-17</td>
<td>09.51.31</td>
</tr>
<tr>
<td>/etc/objrepo</td>
<td>netmgr.clm.obj</td>
<td>SNMP Command Line Manager</td>
<td>09/13/03</td>
<td></td>
<td>2003-01-17</td>
<td>09.51.31</td>
</tr>
<tr>
<td>/etc/objrepo</td>
<td>netmgr.nvdm.obj</td>
<td>NetView Dist. Manager Catcher</td>
<td>09/13/03</td>
<td></td>
<td>2003-01-17</td>
<td>09.51.31</td>
</tr>
<tr>
<td>/etc/objrepo</td>
<td>nv6000.features.ob</td>
<td>AIX SystemView/6000 Fee</td>
<td>09/29/03</td>
<td></td>
<td>2003-01-17</td>
<td>09.51.31</td>
</tr>
<tr>
<td>/etc/objrepo</td>
<td>nv6000.features.ob</td>
<td>AIX SystemView/6000 Base</td>
<td>09/29/03</td>
<td></td>
<td>2003-01-17</td>
<td>09.51.31</td>
</tr>
<tr>
<td>0000</td>
<td>/usr/lib/objrepo</td>
<td>perfmgr</td>
<td>09/10/03</td>
<td>Performance Manager</td>
<td>2003-01-17</td>
<td>09.51.31</td>
</tr>
<tr>
<td>0000</td>
<td>/etc/objrepo</td>
<td>perfagent</td>
<td>09/10/03</td>
<td>Performance Agent</td>
<td>2003-01-17</td>
<td>09.51.31</td>
</tr>
<tr>
<td>0000</td>
<td>/usr/lib/objrepo</td>
<td>perfagent</td>
<td>09/10/03</td>
<td>Performance Agent</td>
<td>2003-01-17</td>
<td>09.51.31</td>
</tr>
<tr>
<td>0000</td>
<td>/usr/lib/objrepo</td>
<td>arc.obj</td>
<td>09/10/03</td>
<td>AIXwindows Interface Composer</td>
<td>2003-01-17</td>
<td>09.51.31</td>
</tr>
<tr>
<td>2937</td>
<td>/usr/lib/objrepo</td>
<td>bsmSv_SE.msg</td>
<td>09/10/03</td>
<td>Base System Messages-Swedish</td>
<td>2003-01-17</td>
<td>09.51.31</td>
</tr>
<tr>
<td>5005</td>
<td>/etc/objrepo</td>
<td>bsmEn_US.msg</td>
<td>09/10/03</td>
<td>Base System Messages-U.S. Eng</td>
<td>2003-01-17</td>
<td>09.51.31</td>
</tr>
<tr>
<td>5005</td>
<td>/etc/objrepo</td>
<td>bsmEn_US.msg</td>
<td>09/10/03</td>
<td>Base System Messages-U.S. Eng</td>
<td>2003-01-17</td>
<td>09.51.31</td>
</tr>
<tr>
<td>5005</td>
<td>/etc/objrepo</td>
<td>bsmEn_US.msg</td>
<td>09/10/03</td>
<td>Base System Messages-U.S. Eng</td>
<td>2003-01-17</td>
<td>09.51.31</td>
</tr>
<tr>
<td>5005</td>
<td>/etc/objrepo</td>
<td>bsmEn_US.msg</td>
<td>09/10/03</td>
<td>Base System Messages-U.S. Eng</td>
<td>2003-01-17</td>
<td>09.51.31</td>
</tr>
<tr>
<td>5005</td>
<td>/etc/objrepo</td>
<td>bsmEn_US.msg</td>
<td>09/10/03</td>
<td>Base System Messages-U.S. Eng</td>
<td>2003-01-17</td>
<td>09.51.31</td>
</tr>
<tr>
<td>5005</td>
<td>/etc/objrepo</td>
<td>bsmEn_US.msg</td>
<td>09/10/03</td>
<td>Base System Messages-U.S. Eng</td>
<td>2003-01-17</td>
<td>09.51.31</td>
</tr>
</tbody>
</table>

Figure 57. Example of UNIX Configuration of SW for a System, 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

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

<table>
<thead>
<tr>
<th>Error type</th>
<th>Errors (count)</th>
<th>Errors (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>9</td>
<td>81.82</td>
</tr>
<tr>
<td>T</td>
<td>2</td>
<td>18.18</td>
</tr>
<tr>
<td>Total</td>
<td>11</td>
<td>100.00</td>
</tr>
</tbody>
</table>

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 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 '

<table>
<thead>
<tr>
<th>Error class (count)</th>
<th>Errors (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>S 10</td>
<td>90.91</td>
</tr>
<tr>
<td>O 1</td>
<td>9.09</td>
</tr>
<tr>
<td>Total 11</td>
<td>100.00</td>
</tr>
</tbody>
</table>

---

**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](image)

**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
Error 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

<table>
<thead>
<tr>
<th>Resource name</th>
<th>Errors 2003-11-01</th>
<th>Errors 2003-12-01</th>
</tr>
</thead>
<tbody>
<tr>
<td>errdemon</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>tok0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>SRC</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>SYSPROC</td>
<td>19</td>
<td>18</td>
</tr>
<tr>
<td>SYSVM</td>
<td>3</td>
<td>5</td>
</tr>
</tbody>
</table>

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 Disk I/O for a Disk, Hourly Trend” on page 178.
- “LINUX Percentile Work Size” on page 182.
- “LINUX Disk Space Allocation” on page 183.
- “LINUX Performance from User Memory” on page 184.
- “LINUX Hardware Configuration” on page 186.
- “LINUX Software Configuration” on page 187.
- “LINUX User Information” on page 188.
- “LINUX Process Information” on page 189.
- “LINUX Performance ‘VM’ for Swap Memory” on page 190.
- “LINUX Performance ‘VM’ for CPU” on page 191.
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

![UNIX Perf CPU Utilization by System, Hourly Trend](image)

*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 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

<table>
<thead>
<tr>
<th>Hour</th>
<th>CPU usage avg (%)</th>
<th>CPU usage max (%)</th>
<th>Mem free avg</th>
<th>Mem free min</th>
<th>Pacing rate avg</th>
<th>Pacing rate max</th>
<th>Processes avg</th>
<th>Processes max</th>
<th>Users avg</th>
<th>Users max</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>6.7</td>
<td>43</td>
<td>1099</td>
<td>166</td>
<td>0.01</td>
<td>0.17</td>
<td>90.3</td>
<td>92</td>
<td>7.0</td>
<td>7</td>
</tr>
<tr>
<td>15</td>
<td>4.8</td>
<td>50</td>
<td>1257</td>
<td>687</td>
<td>0.05</td>
<td>0.12</td>
<td>91.3</td>
<td>96</td>
<td>7.3</td>
<td>12</td>
</tr>
<tr>
<td>16</td>
<td>5.6</td>
<td>21</td>
<td>822</td>
<td>125</td>
<td>0.02</td>
<td>0.09</td>
<td>92.3</td>
<td>99</td>
<td>7.0</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>4.7</td>
<td>50</td>
<td>2447</td>
<td>125</td>
<td>0.06</td>
<td>0.17</td>
<td>85.8</td>
<td>99</td>
<td>6.2</td>
<td>12</td>
</tr>
</tbody>
</table>

Figure 66. Example of UNIX Perf Statistics by 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 pages avg  The average number of free pages of memory.
Performance reports

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
The report contains this information:

<table>
<thead>
<tr>
<th>Node name</th>
<th>CPU usage avg (%)</th>
<th>CPU usage max (%)</th>
<th>Mem free pages avg</th>
<th>Mem free pages min</th>
<th>Paging rate avg</th>
<th>Paging rate max</th>
<th>Processes avg</th>
<th>Processes max</th>
<th>Users avg</th>
<th>Users max</th>
</tr>
</thead>
<tbody>
<tr>
<td>hp111</td>
<td>5.6</td>
<td>50</td>
<td>1068</td>
<td>125</td>
<td>0.01</td>
<td>2</td>
<td>91.4</td>
<td>96</td>
<td>7.1</td>
<td>8</td>
</tr>
<tr>
<td>sun333</td>
<td>18.2</td>
<td>74</td>
<td>621</td>
<td>62</td>
<td>0.15</td>
<td>18</td>
<td>163.9</td>
<td>202</td>
<td>13.5</td>
<td>17</td>
</tr>
</tbody>
</table>

**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:

<table>
<thead>
<tr>
<th>Report ID</th>
<th>Report group</th>
</tr>
</thead>
<tbody>
<tr>
<td>XPERF05</td>
<td>Performance Subcomponent Reports</td>
</tr>
</tbody>
</table>
Performance 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

<table>
<thead>
<tr>
<th>Volume</th>
<th>File system</th>
<th>Space type</th>
<th>Space size avg (MB)</th>
<th>Space free avg (MB)</th>
<th>Space used avg (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>/</td>
<td>/dev/hd4</td>
<td>FS</td>
<td>8.0</td>
<td>0.0</td>
<td>92.0</td>
</tr>
<tr>
<td>/tmp</td>
<td>/dev/hd3</td>
<td>FS</td>
<td>12.0</td>
<td>11.0</td>
<td>6.0</td>
</tr>
<tr>
<td>/usr</td>
<td>/dev/hd2</td>
<td>FS</td>
<td>556.0</td>
<td>67.0</td>
<td>87.0</td>
</tr>
<tr>
<td>/usr/lpp/ileaf6</td>
<td>/dev/lv01</td>
<td>FS</td>
<td>196.0</td>
<td>38.0</td>
<td>80.0</td>
</tr>
<tr>
<td>/var</td>
<td>/dev/hd/var</td>
<td>FS</td>
<td>16.0</td>
<td>5.3</td>
<td>65.0</td>
</tr>
<tr>
<td>rootvg</td>
<td></td>
<td>VG</td>
<td>2212.0</td>
<td>1160.0</td>
<td>47.6</td>
</tr>
</tbody>
</table>

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

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.

---

**Figure 69. Example of UNIX Perf Disk I/O for a Disk, Hourly Trend**

The report contains this information:

<table>
<thead>
<tr>
<th>Hour</th>
<th>Busy avg (%)</th>
<th>Busy max (%)</th>
<th>Read avg (KB/sec)</th>
<th>Read max (KB/sec)</th>
<th>Write avg (KB/sec)</th>
<th>Write max (KB/sec)</th>
<th>RW avg (KB/sec)</th>
<th>RW max (KB/sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>12.9</td>
<td>12.0</td>
<td>130.6</td>
<td>52.0</td>
<td>5.4</td>
<td>18.0</td>
<td>7.1</td>
<td>70.0</td>
</tr>
<tr>
<td>13</td>
<td>9.9</td>
<td>40.0</td>
<td>14.0</td>
<td>4.3</td>
<td>8.0</td>
<td>4.7</td>
<td>22.0</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>6.6</td>
<td>24.2</td>
<td>0.1</td>
<td>4.4</td>
<td>8.0</td>
<td>4.5</td>
<td>14.0</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>53.1</td>
<td>361.1</td>
<td>20.0</td>
<td>11.3</td>
<td>67.0</td>
<td>31.3</td>
<td>286.0</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>16.3</td>
<td>87.2</td>
<td>3.7</td>
<td>5.1</td>
<td>13.0</td>
<td>8.8</td>
<td>48.0</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>17.3</td>
<td>155.3</td>
<td>3.2</td>
<td>6.2</td>
<td>14.0</td>
<td>9.4</td>
<td>69.0</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>8.1</td>
<td>19.0</td>
<td>0.2</td>
<td>5.0</td>
<td>4.1</td>
<td>6.0</td>
<td>4.3</td>
<td>11.0</td>
</tr>
<tr>
<td>19</td>
<td>7.6</td>
<td>10.0</td>
<td>1.0</td>
<td>10.2</td>
<td>3.6</td>
<td>5.0</td>
<td>4.6</td>
<td>15.2</td>
</tr>
<tr>
<td>20</td>
<td>7.7</td>
<td>9.7</td>
<td>0.5</td>
<td>4.1</td>
<td>4.9</td>
<td>5.0</td>
<td>5.4</td>
<td>9.1</td>
</tr>
<tr>
<td>21</td>
<td>7.7</td>
<td>10.6</td>
<td>1.5</td>
<td>17.3</td>
<td>5.0</td>
<td>4.0</td>
<td>5.5</td>
<td>21.3</td>
</tr>
<tr>
<td>22</td>
<td>7.7</td>
<td>9.8</td>
<td>0.1</td>
<td>1.9</td>
<td>2.2</td>
<td>5.0</td>
<td>2.3</td>
<td>6.9</td>
</tr>
<tr>
<td>23</td>
<td>7.6</td>
<td>9.8</td>
<td>0.3</td>
<td>2.2</td>
<td>4.5</td>
<td>5.0</td>
<td>4.8</td>
<td>7.2</td>
</tr>
</tbody>
</table>

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
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

---

<table>
<thead>
<tr>
<th>Disk name</th>
<th>Busy avg (%)</th>
<th>Busy max (%)</th>
<th>Read avg (KB/sec)</th>
<th>Read max (KB/sec)</th>
<th>Write avg (KB/sec)</th>
<th>Write max (KB/sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>hdisk0</td>
<td>33.5</td>
<td>17.0</td>
<td>1.0</td>
<td>1.4</td>
<td>11.1</td>
<td>40.0</td>
</tr>
<tr>
<td>hdisk1</td>
<td>11.4</td>
<td>14.5</td>
<td>1.1</td>
<td>1.4</td>
<td>10.4</td>
<td>12.2</td>
</tr>
<tr>
<td>hdisk2</td>
<td>10.0</td>
<td>19.1</td>
<td>1.0</td>
<td>1.0</td>
<td>10.0</td>
<td>1.2</td>
</tr>
<tr>
<td>hdisk3</td>
<td>18.3</td>
<td>11.1</td>
<td>1.5</td>
<td>11.3</td>
<td>11.1</td>
<td>18.0</td>
</tr>
<tr>
<td><strong>Avg/max</strong></td>
<td><strong>17.3</strong></td>
<td><strong>19.1</strong></td>
<td><strong>1.1</strong></td>
<td><strong>13.0</strong></td>
<td><strong>10.6</strong></td>
<td><strong>40.0</strong></td>
</tr>
</tbody>
</table>

---

*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
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

<table>
<thead>
<tr>
<th>DATE</th>
<th>PERIOD</th>
<th>FILE SYS</th>
<th>MAX SIZE</th>
<th>% WORK SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2001-01-16</td>
<td>PRIME</td>
<td>/dev/hda7</td>
<td>23474136</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>/dev/hda1</td>
<td>54416</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>/dev/hda5</td>
<td>5044156</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>NIGHT</td>
<td>/dev/hda7</td>
<td>164318952</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>/dev/hda1</td>
<td>380912</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>/dev/hda5</td>
<td>35309092</td>
<td>6</td>
</tr>
</tbody>
</table>

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
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
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
Performance reports

<table>
<thead>
<tr>
<th>DATE</th>
<th>PERIOD</th>
<th>VOLUME</th>
<th>SIZE</th>
<th>BLOCKS</th>
</tr>
</thead>
<tbody>
<tr>
<td>2001-01-16</td>
<td>PRIME</td>
<td>/dev/hda6</td>
<td>2.650E+05</td>
<td>5.360E+02</td>
</tr>
<tr>
<td>2001-01-16</td>
<td>NIGHT</td>
<td>/dev/hda6</td>
<td>2.650E+05</td>
<td>5.360E+02</td>
</tr>
</tbody>
</table>

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
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

<table>
<thead>
<tr>
<th>DATE</th>
<th>PERIOD</th>
<th>PACKAGE</th>
<th>VERSION</th>
</tr>
</thead>
<tbody>
<tr>
<td>2001-01-16</td>
<td>NIGHT</td>
<td>anacron</td>
<td>2.1</td>
</tr>
<tr>
<td>2001-01-16</td>
<td>NIGHT</td>
<td>apmd</td>
<td>3.0</td>
</tr>
<tr>
<td>2001-01-16</td>
<td>NIGHT</td>
<td>ash</td>
<td>0.2</td>
</tr>
<tr>
<td>2001-01-16</td>
<td>NIGHT</td>
<td>at</td>
<td>3.1</td>
</tr>
<tr>
<td>2001-01-16</td>
<td>NIGHT</td>
<td>audiofile</td>
<td>0.1</td>
</tr>
<tr>
<td>2001-01-16</td>
<td>NIGHT</td>
<td>audiofile-devel</td>
<td>0.1</td>
</tr>
<tr>
<td>2001-01-16</td>
<td>NIGHT</td>
<td>auix</td>
<td>1.3</td>
</tr>
<tr>
<td>2001-01-16</td>
<td>NIGHT</td>
<td>authconfig</td>
<td>3.0</td>
</tr>
<tr>
<td>2001-01-16</td>
<td>NIGHT</td>
<td>authconf</td>
<td>2.1</td>
</tr>
</tbody>
</table>

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
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
Performance reports

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
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
Performance reports

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 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 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

---

*Figure 81. Example of a LINUX - Performance 'VM' for CPU report*
Windows component reports

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'*

<table>
<thead>
<tr>
<th>NODE NAME</th>
<th>DEVICE NAME</th>
<th>TIME</th>
<th>FREE SPACE MIN (MB)</th>
<th>FREE SPACE AVG (MB)</th>
<th>FREE SPACE MAX (MB)</th>
<th>PERCENT FREE MIN</th>
<th>PERCENT FREE AVG</th>
<th>PERCENT FREE MAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM-PAXPRIM</td>
<td>2009-06-12</td>
<td>C:</td>
<td>00.00.00</td>
<td>61691</td>
<td>61691</td>
<td>61691</td>
<td>67</td>
<td>67</td>
</tr>
<tr>
<td></td>
<td></td>
<td>C:</td>
<td>01.00.00</td>
<td>61684</td>
<td>61688</td>
<td>61691</td>
<td>67</td>
<td>67</td>
</tr>
<tr>
<td></td>
<td></td>
<td>C:</td>
<td>02.00.00</td>
<td>61684</td>
<td>61684</td>
<td>61684</td>
<td>67</td>
<td>67</td>
</tr>
<tr>
<td></td>
<td></td>
<td>C:</td>
<td>03.00.00</td>
<td>61684</td>
<td>61684</td>
<td>61684</td>
<td>67</td>
<td>67</td>
</tr>
<tr>
<td></td>
<td></td>
<td>C:</td>
<td>04.00.00</td>
<td>61684</td>
<td>61684</td>
<td>61684</td>
<td>67</td>
<td>67</td>
</tr>
<tr>
<td></td>
<td></td>
<td>C:</td>
<td>05.00.00</td>
<td>61684</td>
<td>61702</td>
<td>61720</td>
<td>67</td>
<td>67</td>
</tr>
<tr>
<td></td>
<td></td>
<td>C:</td>
<td>06.00.00</td>
<td>61720</td>
<td>61720</td>
<td>61720</td>
<td>67</td>
<td>67</td>
</tr>
<tr>
<td></td>
<td></td>
<td>C:</td>
<td>07.00.00</td>
<td>61720</td>
<td>61720</td>
<td>61720</td>
<td>67</td>
<td>67</td>
</tr>
<tr>
<td></td>
<td></td>
<td>C:</td>
<td>08.00.00</td>
<td>61720</td>
<td>61720</td>
<td>61720</td>
<td>67</td>
<td>67</td>
</tr>
<tr>
<td></td>
<td></td>
<td>C:</td>
<td>09.00.00</td>
<td>61720</td>
<td>61720</td>
<td>61720</td>
<td>67</td>
<td>67</td>
</tr>
<tr>
<td></td>
<td></td>
<td>C:</td>
<td>10.00.00</td>
<td>61594</td>
<td>61595</td>
<td>61595</td>
<td>67</td>
<td>67</td>
</tr>
<tr>
<td></td>
<td></td>
<td>C:</td>
<td>11.00.00</td>
<td>61587</td>
<td>61588</td>
<td>61588</td>
<td>67</td>
<td>67</td>
</tr>
<tr>
<td></td>
<td></td>
<td>C:</td>
<td>12.00.00</td>
<td>61586</td>
<td>61587</td>
<td>61587</td>
<td>67</td>
<td>67</td>
</tr>
<tr>
<td></td>
<td></td>
<td>C:</td>
<td>13.00.00</td>
<td>61584</td>
<td>61586</td>
<td>61586</td>
<td>67</td>
<td>67</td>
</tr>
<tr>
<td></td>
<td></td>
<td>C:</td>
<td>14.00.00</td>
<td>61565</td>
<td>61565</td>
<td>61564</td>
<td>67</td>
<td>67</td>
</tr>
<tr>
<td></td>
<td></td>
<td>C:</td>
<td>15.00.00</td>
<td>61634</td>
<td>61635</td>
<td>61635</td>
<td>67</td>
<td>67</td>
</tr>
<tr>
<td></td>
<td></td>
<td>C:</td>
<td>16.00.00</td>
<td>61631</td>
<td>61633</td>
<td>61634</td>
<td>67</td>
<td>67</td>
</tr>
<tr>
<td></td>
<td></td>
<td>C:</td>
<td>17.00.00</td>
<td>61629</td>
<td>61633</td>
<td>61636</td>
<td>67</td>
<td>67</td>
</tr>
<tr>
<td></td>
<td></td>
<td>C:</td>
<td>18.00.00</td>
<td>61635</td>
<td>61635</td>
<td>61635</td>
<td>67</td>
<td>67</td>
</tr>
<tr>
<td></td>
<td></td>
<td>C:</td>
<td>19.00.00</td>
<td>61633</td>
<td>61634</td>
<td>61634</td>
<td>67</td>
<td>67</td>
</tr>
<tr>
<td></td>
<td></td>
<td>C:</td>
<td>20.00.00</td>
<td>61632</td>
<td>61632</td>
<td>61632</td>
<td>67</td>
<td>67</td>
</tr>
<tr>
<td></td>
<td></td>
<td>C:</td>
<td>21.00.00</td>
<td>61631</td>
<td>61632</td>
<td>61632</td>
<td>67</td>
<td>67</td>
</tr>
<tr>
<td></td>
<td></td>
<td>C:</td>
<td>22.00.00</td>
<td>61630</td>
<td>61631</td>
<td>61631</td>
<td>67</td>
<td>67</td>
</tr>
<tr>
<td></td>
<td></td>
<td>C:</td>
<td>23.00.00</td>
<td>61629</td>
<td>61630</td>
<td>61630</td>
<td>67</td>
<td>67</td>
</tr>
</tbody>
</table>

---

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 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, 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 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, 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

<table>
<thead>
<tr>
<th>NODE NAME</th>
<th>DATE</th>
<th>TIME</th>
<th>MEMORY FREE MIN (KB)</th>
<th>MEMORY FREE AVG (KB)</th>
<th>MEMORY FREE MAX (KB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM-PAXPRIM</td>
<td>2009-06-12 00.00.00</td>
<td>1399924</td>
<td>1407371</td>
<td>1408564</td>
<td></td>
</tr>
<tr>
<td></td>
<td>01.00.00</td>
<td>1364320</td>
<td>1367920</td>
<td>1368132</td>
<td></td>
</tr>
<tr>
<td></td>
<td>02.00.00</td>
<td>1367096</td>
<td>1376724</td>
<td>1379888</td>
<td></td>
</tr>
<tr>
<td></td>
<td>03.00.00</td>
<td>1366520</td>
<td>1367241</td>
<td>1367988</td>
<td></td>
</tr>
<tr>
<td></td>
<td>04.00.00</td>
<td>1366172</td>
<td>1366796</td>
<td>1367520</td>
<td></td>
</tr>
<tr>
<td></td>
<td>05.00.00</td>
<td>1346256</td>
<td>1380571</td>
<td>1389388</td>
<td></td>
</tr>
<tr>
<td></td>
<td>06.00.00</td>
<td>1387428</td>
<td>1388094</td>
<td>1388876</td>
<td></td>
</tr>
<tr>
<td></td>
<td>07.00.00</td>
<td>1376300</td>
<td>1387421</td>
<td>1388156</td>
<td></td>
</tr>
<tr>
<td></td>
<td>08.00.00</td>
<td>1386156</td>
<td>1386948</td>
<td>1387664</td>
<td></td>
</tr>
<tr>
<td></td>
<td>09.00.00</td>
<td>1015916</td>
<td>1353313</td>
<td>1399148</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10.00.00</td>
<td>1045336</td>
<td>1155013</td>
<td>1214188</td>
<td></td>
</tr>
<tr>
<td></td>
<td>11.00.00</td>
<td>1015476</td>
<td>1056817</td>
<td>1120244</td>
<td></td>
</tr>
<tr>
<td></td>
<td>12.00.00</td>
<td>1100632</td>
<td>1152908</td>
<td>1213140</td>
<td></td>
</tr>
<tr>
<td></td>
<td>13.00.00</td>
<td>1062504</td>
<td>1136608</td>
<td>1200420</td>
<td></td>
</tr>
<tr>
<td></td>
<td>14.00.00</td>
<td>920532</td>
<td>984165</td>
<td>1096668</td>
<td></td>
</tr>
<tr>
<td></td>
<td>15.00.00</td>
<td>893760</td>
<td>988563</td>
<td>1047896</td>
<td></td>
</tr>
<tr>
<td></td>
<td>16.00.00</td>
<td>924324</td>
<td>983061</td>
<td>1050484</td>
<td></td>
</tr>
<tr>
<td></td>
<td>17.00.00</td>
<td>937800</td>
<td>958081</td>
<td>963300</td>
<td></td>
</tr>
<tr>
<td></td>
<td>18.00.00</td>
<td>928756</td>
<td>941137</td>
<td>944116</td>
<td></td>
</tr>
<tr>
<td></td>
<td>19.00.00</td>
<td>938352</td>
<td>940428</td>
<td>941956</td>
<td></td>
</tr>
<tr>
<td></td>
<td>20.00.00</td>
<td>934356</td>
<td>939315</td>
<td>940936</td>
<td></td>
</tr>
<tr>
<td></td>
<td>21.00.00</td>
<td>919056</td>
<td>937112</td>
<td>939584</td>
<td></td>
</tr>
<tr>
<td></td>
<td>22.00.00</td>
<td>928324</td>
<td>933017</td>
<td>935268</td>
<td></td>
</tr>
<tr>
<td></td>
<td>23.00.00</td>
<td>925836</td>
<td>932618</td>
<td>933872</td>
<td></td>
</tr>
</tbody>
</table>

Figure 86. Example of Windows Memory Usage by System, Hourly Trend
Windows component reports

Windows Memory Usage by System, Daily Overview

<table>
<thead>
<tr>
<th>NODE NAME</th>
<th>DATE</th>
<th>MEMORY_MIN (KB)</th>
<th>MEMORY_AVG (KB)</th>
<th>MEMORY_MAX (KB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM-PAXPRIM</td>
<td>2009-06-12</td>
<td>893760</td>
<td>1159459</td>
<td>1408564</td>
</tr>
</tbody>
</table>

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
<table>
<thead>
<tr>
<th>TIME</th>
<th>CPU USAGE</th>
<th>CPU USAGE</th>
<th>MEMORY FREE</th>
<th>MEMORY FREE</th>
<th>FREE SPACE</th>
<th>FREE SPACE</th>
<th>FREE SPACE</th>
<th>FREE SPACE</th>
</tr>
</thead>
<tbody>
<tr>
<td>01.00.00</td>
<td>0 0 0</td>
<td>761436</td>
<td>767177</td>
<td>769700</td>
<td>6 6</td>
<td>6</td>
<td>71059</td>
<td>71059</td>
</tr>
<tr>
<td>02.00.00</td>
<td>0 0 0</td>
<td>762992</td>
<td>767326</td>
<td>768264</td>
<td>6 6</td>
<td>6</td>
<td>71059</td>
<td>71059</td>
</tr>
<tr>
<td>03.00.00</td>
<td>0 0 0</td>
<td>762688</td>
<td>767188</td>
<td>769344</td>
<td>6 6</td>
<td>6</td>
<td>71059</td>
<td>71059</td>
</tr>
<tr>
<td>04.00.00</td>
<td>0 0 0</td>
<td>763248</td>
<td>767604</td>
<td>769248</td>
<td>6 6</td>
<td>6</td>
<td>71059</td>
<td>71059</td>
</tr>
<tr>
<td>05.00.00</td>
<td>0 0 0</td>
<td>762536</td>
<td>767026</td>
<td>767936</td>
<td>6 6</td>
<td>6</td>
<td>71059</td>
<td>71059</td>
</tr>
<tr>
<td>06.00.00</td>
<td>0 0 0</td>
<td>750628</td>
<td>765030</td>
<td>768428</td>
<td>6 6</td>
<td>6</td>
<td>71059</td>
<td>71059</td>
</tr>
<tr>
<td>07.00.00</td>
<td>0 0 0</td>
<td>761988</td>
<td>768684</td>
<td>768684</td>
<td>6 6</td>
<td>6</td>
<td>71059</td>
<td>71059</td>
</tr>
<tr>
<td>08.00.00</td>
<td>0 0 0</td>
<td>761828</td>
<td>766428</td>
<td>767796</td>
<td>6 6</td>
<td>6</td>
<td>71059</td>
<td>71059</td>
</tr>
<tr>
<td>09.00.00</td>
<td>0 0 0</td>
<td>762660</td>
<td>766984</td>
<td>768732</td>
<td>6 6</td>
<td>6</td>
<td>71059</td>
<td>71059</td>
</tr>
<tr>
<td>10.00.00</td>
<td>0 0 0</td>
<td>761768</td>
<td>766314</td>
<td>767748</td>
<td>6 6</td>
<td>6</td>
<td>71059</td>
<td>71059</td>
</tr>
<tr>
<td>11.00.00</td>
<td>0 0 0</td>
<td>761848</td>
<td>767150</td>
<td>769156</td>
<td>6 6</td>
<td>6</td>
<td>71059</td>
<td>71059</td>
</tr>
<tr>
<td>12.00.00</td>
<td>0 0 0</td>
<td>763076</td>
<td>767396</td>
<td>769036</td>
<td>6 6</td>
<td>6</td>
<td>71059</td>
<td>71059</td>
</tr>
<tr>
<td>13.00.00</td>
<td>0 0 0</td>
<td>761764</td>
<td>766653</td>
<td>768648</td>
<td>6 6</td>
<td>6</td>
<td>71059</td>
<td>71059</td>
</tr>
<tr>
<td>14.00.00</td>
<td>0 0 0</td>
<td>762248</td>
<td>769011</td>
<td>768456</td>
<td>6 6</td>
<td>6</td>
<td>71059</td>
<td>71059</td>
</tr>
<tr>
<td>15.00.00</td>
<td>0 0 0</td>
<td>761992</td>
<td>766368</td>
<td>767444</td>
<td>6 6</td>
<td>6</td>
<td>71059</td>
<td>71059</td>
</tr>
<tr>
<td>16.00.00</td>
<td>0 0 0</td>
<td>759528</td>
<td>765585</td>
<td>767356</td>
<td>6 6</td>
<td>6</td>
<td>71059</td>
<td>71059</td>
</tr>
<tr>
<td>17.00.00</td>
<td>0 0 0</td>
<td>761256</td>
<td>765821</td>
<td>767588</td>
<td>6 6</td>
<td>6</td>
<td>71059</td>
<td>71059</td>
</tr>
<tr>
<td>18.00.00</td>
<td>0 0 0</td>
<td>760948</td>
<td>765652</td>
<td>767776</td>
<td>6 6</td>
<td>6</td>
<td>71059</td>
<td>71059</td>
</tr>
<tr>
<td>19.00.00</td>
<td>0 0 0</td>
<td>761504</td>
<td>765825</td>
<td>767204</td>
<td>6 6</td>
<td>6</td>
<td>71059</td>
<td>71059</td>
</tr>
<tr>
<td>20.00.00</td>
<td>0 0 0</td>
<td>760660</td>
<td>765440</td>
<td>767024</td>
<td>6 6</td>
<td>6</td>
<td>71059</td>
<td>71059</td>
</tr>
<tr>
<td>21.00.00</td>
<td>0 0 0</td>
<td>760556</td>
<td>765945</td>
<td>767820</td>
<td>6 6</td>
<td>6</td>
<td>71059</td>
<td>71059</td>
</tr>
<tr>
<td>22.00.00</td>
<td>0 0 0</td>
<td>761356</td>
<td>765693</td>
<td>767436</td>
<td>6 6</td>
<td>6</td>
<td>71059</td>
<td>71059</td>
</tr>
<tr>
<td>23.00.00</td>
<td>0 0 0</td>
<td>760828</td>
<td>766078</td>
<td>768404</td>
<td>6 6</td>
<td>6</td>
<td>71059</td>
<td>71059</td>
</tr>
</tbody>
</table>

**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” on page 203
- “XACCT_CONNECT” on page 202
- “XACCT_DISK” on page 202
- “XACCT_PRINT” on page 203
- “XCONFIG_HW” on page 203
- “XCONFIG_SW” on page 203
- “XERROR” on page 205
- “XPERF_CPU” on page 206
- “XPERF_DISK_FS” on page 206
- “XPERF_DISK_VG” on page 207
- “XPERF_DISKIO” on page 208
- “XPERF_PAGING” on page 209
- “WIN_PERF_DD” on page 209
- “WIN_PERF_PU” on page 210

XACCT_COMMAND

This process accounting record definition is used for selecting records with record type PACCT.

Main section

<table>
<thead>
<tr>
<th>Field name</th>
<th>Offset</th>
<th>Length</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RECORD_LENGTH</td>
<td>0</td>
<td>2</td>
<td>BINARY</td>
<td>Record length</td>
</tr>
<tr>
<td>NODE_NAME</td>
<td>4</td>
<td>16</td>
<td>CHAR</td>
<td>Node name</td>
</tr>
<tr>
<td>RECORD_TYPE</td>
<td>21</td>
<td>5</td>
<td>CHAR</td>
<td>Record type</td>
</tr>
<tr>
<td>DTE</td>
<td>27</td>
<td>6</td>
<td>CHAR</td>
<td>Date</td>
</tr>
<tr>
<td>TME</td>
<td>33</td>
<td>6</td>
<td>TIME(HHMMSS)</td>
<td>Time</td>
</tr>
<tr>
<td>AC_FLAG</td>
<td>40</td>
<td>5</td>
<td>EXTERNAL INTEGER</td>
<td>Acct. flag</td>
</tr>
<tr>
<td>AC_STAT</td>
<td>46</td>
<td>5</td>
<td>EXTERNAL INTEGER</td>
<td>Exit status</td>
</tr>
<tr>
<td>AC_UID</td>
<td>52</td>
<td>8</td>
<td>CHAR</td>
<td>User name</td>
</tr>
<tr>
<td>AC_GID</td>
<td>61</td>
<td>8</td>
<td>CHAR</td>
<td>Group name</td>
</tr>
<tr>
<td>AC_TTY</td>
<td>70</td>
<td>10</td>
<td>CHAR</td>
<td>Control typewriter</td>
</tr>
<tr>
<td>AC_COMM</td>
<td>81</td>
<td>12</td>
<td>CHAR</td>
<td>Command name</td>
</tr>
<tr>
<td>AC_UTIME</td>
<td>94</td>
<td>10</td>
<td>EXTERNAL FLOAT</td>
<td>User time</td>
</tr>
<tr>
<td>AC_STIME</td>
<td>105</td>
<td>10</td>
<td>EXTERNAL FLOAT</td>
<td>System time</td>
</tr>
<tr>
<td>ACETIME</td>
<td>116</td>
<td>16</td>
<td>EXTERNAL FLOAT</td>
<td>Elapsed time</td>
</tr>
<tr>
<td>AC_MEM</td>
<td>133</td>
<td>10</td>
<td>EXTERNAL FLOAT</td>
<td>Memory usage</td>
</tr>
<tr>
<td>AC_IO</td>
<td>144</td>
<td>16</td>
<td>EXTERNAL FLOAT</td>
<td>Characters Transferred</td>
</tr>
<tr>
<td>AC_RW</td>
<td>161</td>
<td>10</td>
<td>EXTERNAL FLOAT</td>
<td>Blocks Read/Written</td>
</tr>
</tbody>
</table>
**XACCT_CONNECT**

This connect time accounting record definition is used for selecting records with record type WTMP.

**Main section**

<table>
<thead>
<tr>
<th>Field name</th>
<th>Offset</th>
<th>Length</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RECORD_LENGTH</td>
<td>0</td>
<td>2</td>
<td>BINARY</td>
<td>Record length</td>
</tr>
<tr>
<td>NODE_NAME</td>
<td>4</td>
<td>16</td>
<td>CHAR</td>
<td>Node name</td>
</tr>
<tr>
<td>RECORD_TYPE</td>
<td>21</td>
<td>5</td>
<td>CHAR</td>
<td>Record type</td>
</tr>
<tr>
<td>DTE</td>
<td>27</td>
<td>6</td>
<td>CHAR</td>
<td>Date</td>
</tr>
<tr>
<td>TME</td>
<td>33</td>
<td>6</td>
<td>TIME(HHMMSS)</td>
<td>Time</td>
</tr>
<tr>
<td>UT_USER</td>
<td>40</td>
<td>8</td>
<td>CHAR</td>
<td>User login name</td>
</tr>
<tr>
<td>UT_ID</td>
<td>49</td>
<td>14</td>
<td>CHAR</td>
<td>/etc/inittab ID</td>
</tr>
<tr>
<td>UT_LINE</td>
<td>64</td>
<td>12</td>
<td>CHAR</td>
<td>Device name</td>
</tr>
<tr>
<td>UT_TYPE</td>
<td>77</td>
<td>2</td>
<td>CHAR</td>
<td>Type of entry</td>
</tr>
<tr>
<td>UT_PID</td>
<td>80</td>
<td>6</td>
<td>EXTERNAL INTEGER</td>
<td>Connect ID</td>
</tr>
<tr>
<td>E_TERMINATION</td>
<td>87</td>
<td>2</td>
<td>CHAR</td>
<td>Connect term status</td>
</tr>
<tr>
<td>E_EXIT</td>
<td>90</td>
<td>2</td>
<td>CHAR</td>
<td>Connect exit status</td>
</tr>
<tr>
<td>UT_TIME</td>
<td>93</td>
<td>10</td>
<td>EXTERNAL FLOAT</td>
<td>Time entry was made (see Note)</td>
</tr>
<tr>
<td>UT_HOST</td>
<td>104</td>
<td>*</td>
<td>CHAR</td>
<td>Node name</td>
</tr>
</tbody>
</table>

**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**

<table>
<thead>
<tr>
<th>Field name</th>
<th>Offset</th>
<th>Length</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RECORD_LENGTH</td>
<td>0</td>
<td>2</td>
<td>BINARY</td>
<td>Record length</td>
</tr>
<tr>
<td>NODE_NAME</td>
<td>4</td>
<td>16</td>
<td>CHAR</td>
<td>Node name</td>
</tr>
<tr>
<td>RECORD_TYPE</td>
<td>21</td>
<td>5</td>
<td>CHAR</td>
<td>Record type</td>
</tr>
<tr>
<td>DTE</td>
<td>27</td>
<td>6</td>
<td>CHAR</td>
<td>Date</td>
</tr>
<tr>
<td>TME</td>
<td>33</td>
<td>6</td>
<td>TIME(HHMMSS)</td>
<td>Time</td>
</tr>
<tr>
<td>USERID</td>
<td>40</td>
<td>11</td>
<td>EXTERNAL INTEGER</td>
<td>User ID number</td>
</tr>
<tr>
<td>USER_NAME</td>
<td>52</td>
<td>8</td>
<td>CHAR</td>
<td>User name</td>
</tr>
<tr>
<td>DISK_BLOCKS</td>
<td>61</td>
<td>8</td>
<td>EXTERNAL INTEGER</td>
<td>Disk blocks</td>
</tr>
</tbody>
</table>
XACCT_PRINT

This print accounting record definition is used for selecting records with record type QACCT.

Main section

<table>
<thead>
<tr>
<th>Field name</th>
<th>Offset</th>
<th>Length</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RECORD_LENGTH</td>
<td>0</td>
<td>2</td>
<td>BINAR</td>
<td>Record length</td>
</tr>
<tr>
<td>NODE_NAME</td>
<td>4</td>
<td>16</td>
<td>CHAR</td>
<td>Node name</td>
</tr>
<tr>
<td>RECORD_TYPE</td>
<td>21</td>
<td>5</td>
<td>CHAR</td>
<td>Record type</td>
</tr>
<tr>
<td>DTE</td>
<td>27</td>
<td>6</td>
<td>CHAR</td>
<td>Date</td>
</tr>
<tr>
<td>TME</td>
<td>33</td>
<td>6</td>
<td>TIME(HHMMSS)</td>
<td>Time</td>
</tr>
<tr>
<td>PRINT_QUEUE</td>
<td>40</td>
<td>8</td>
<td>CHAR</td>
<td>Printer queue remote name</td>
</tr>
<tr>
<td>PAGES</td>
<td>49</td>
<td>8</td>
<td>EXTERNAL INTEGER</td>
<td>Printed pages</td>
</tr>
<tr>
<td>COPIES</td>
<td>58</td>
<td>8</td>
<td>EXTERNAL INTEGER</td>
<td>The number of printouts</td>
</tr>
<tr>
<td>PR_USER</td>
<td>67</td>
<td>*</td>
<td>CHAR(*)</td>
<td>User and requesting node</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Field name</th>
<th>Offset</th>
<th>Length</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RECORD_LENGTH</td>
<td>0</td>
<td>2</td>
<td>BINAR</td>
<td>Record length</td>
</tr>
<tr>
<td>NODE_NAME</td>
<td>4</td>
<td>16</td>
<td>CHAR</td>
<td>Node name</td>
</tr>
<tr>
<td>RECORD_TYPE</td>
<td>21</td>
<td>5</td>
<td>CHAR</td>
<td>Record type</td>
</tr>
<tr>
<td>DTE</td>
<td>27</td>
<td>6</td>
<td>CHAR</td>
<td>Date</td>
</tr>
<tr>
<td>TME</td>
<td>33</td>
<td>6</td>
<td>TIME(HHMMSS)</td>
<td>Time</td>
</tr>
<tr>
<td>RECORD_SUBTYPE</td>
<td>40</td>
<td>2</td>
<td>CHAR</td>
<td>Record subtype</td>
</tr>
<tr>
<td>DEVICE_NAME</td>
<td>43</td>
<td>16</td>
<td>CHAR</td>
<td>Device name</td>
</tr>
<tr>
<td>LOCATION</td>
<td>60</td>
<td>16</td>
<td>CHAR</td>
<td>Location</td>
</tr>
<tr>
<td>DEVICE_CLASS</td>
<td>77</td>
<td>16</td>
<td>CHAR</td>
<td>Device class</td>
</tr>
<tr>
<td>DEVICE_TYPE</td>
<td>94</td>
<td>16</td>
<td>CHAR</td>
<td>Device type</td>
</tr>
<tr>
<td>DEVICE_SUBCLASS</td>
<td>111</td>
<td>16</td>
<td>CHAR</td>
<td>Device subclass</td>
</tr>
<tr>
<td>DEVICE_DESC</td>
<td>128</td>
<td>*</td>
<td>CHAR(*)</td>
<td>Device descr.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Field name</th>
<th>Offset</th>
<th>Length</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RECORD_LENGTH</td>
<td>0</td>
<td>2</td>
<td>BINARY</td>
<td>Record length</td>
</tr>
<tr>
<td>NODE_NAME</td>
<td>4</td>
<td>16</td>
<td>CHAR</td>
<td>Node name</td>
</tr>
<tr>
<td>RECORD_TYPE</td>
<td>21</td>
<td>5</td>
<td>CHAR</td>
<td>Record type</td>
</tr>
<tr>
<td>DTE</td>
<td>27</td>
<td>6</td>
<td>CHAR</td>
<td>Date</td>
</tr>
<tr>
<td>TME</td>
<td>33</td>
<td>6</td>
<td>TIME(HHMMSS)</td>
<td>Time</td>
</tr>
<tr>
<td>RECORD_SUBTYPE</td>
<td>40</td>
<td>2</td>
<td>CHAR</td>
<td>Record subtype</td>
</tr>
</tbody>
</table>

## Record section SUBTYPE_H

**About this task**

Present if RECORD_SUBTYPE has the value H.

**Offset** 42  
**Length** 36

<table>
<thead>
<tr>
<th>Field name</th>
<th>Offset</th>
<th>Length</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RELEASE</td>
<td>1</td>
<td>15</td>
<td>CHAR</td>
<td>Release number</td>
</tr>
<tr>
<td>STATUS</td>
<td>17</td>
<td>10</td>
<td>CHAR</td>
<td>Installation status</td>
</tr>
<tr>
<td>INST_DATE</td>
<td>28</td>
<td>8</td>
<td>CHAR</td>
<td>Installation date</td>
</tr>
</tbody>
</table>

## Record section SUBTYPE_I

Present if RECORD_SUBTYPE has the value I.

**Offset** 42  
**Length** 37

<table>
<thead>
<tr>
<th>Field name</th>
<th>Offset</th>
<th>Length</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRODUCT_ID</td>
<td>1</td>
<td>10</td>
<td>CHAR</td>
<td>Product ID</td>
</tr>
<tr>
<td>FEATURE_ID</td>
<td>12</td>
<td>4</td>
<td>CHAR</td>
<td>Feature ID</td>
</tr>
<tr>
<td>PRODUCT_NAME</td>
<td>17</td>
<td>20</td>
<td>CHAR</td>
<td>Product name</td>
</tr>
</tbody>
</table>

## Record section SUBTYPE_L

Present if RECORD_SUBTYPE has the value L.

**Offset** 42  
**Length** 13

<table>
<thead>
<tr>
<th>Field name</th>
<th>Offset</th>
<th>Length</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STATE</td>
<td>1</td>
<td>12</td>
<td>CHAR</td>
<td>Product state</td>
</tr>
</tbody>
</table>

## Record section FILE_NAME

**Offset** 79  
**Length** Length of FILE_NAME_L
Record section PATH_NAME
Offset  Dependent upon length of previous section
Length  Length of PATH_NAME_L

<table>
<thead>
<tr>
<th>Field name</th>
<th>Offset</th>
<th>Length</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PATH_NAME_L</td>
<td>1</td>
<td>4</td>
<td>EXTERNAL INTEGER</td>
<td>Length of section PATH_NAME</td>
</tr>
<tr>
<td>PATH_NAME</td>
<td>6</td>
<td>*</td>
<td>CHAR(*)</td>
<td>Path name</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Field name</th>
<th>Offset</th>
<th>Length</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRODUCT_DESC_L</td>
<td>1</td>
<td>4</td>
<td>EXTERNAL INTEGER</td>
<td>Length of section PRODUCT_DESC</td>
</tr>
<tr>
<td>PRODUCT_DESC</td>
<td>6</td>
<td>*</td>
<td>CHAR(*)</td>
<td>Product description</td>
</tr>
</tbody>
</table>

XERROR
This error record definition is used for selecting records with record type ERRPT.

Main section

<table>
<thead>
<tr>
<th>Field name</th>
<th>Offset</th>
<th>Length</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RECORD_LENGTH</td>
<td>0</td>
<td>2</td>
<td>BINARY</td>
<td>Record length</td>
</tr>
<tr>
<td>NODE_NAME</td>
<td>4</td>
<td>16</td>
<td>CHAR</td>
<td>Node name</td>
</tr>
<tr>
<td>RECORD_TYPE</td>
<td>21</td>
<td>5</td>
<td>CHAR</td>
<td>Record type</td>
</tr>
<tr>
<td>DTE</td>
<td>27</td>
<td>6</td>
<td>CHAR</td>
<td>Date</td>
</tr>
<tr>
<td>TME</td>
<td>33</td>
<td>6</td>
<td>TIME(HHMMSS)</td>
<td>Time</td>
</tr>
<tr>
<td>ERROR_ID</td>
<td>40</td>
<td>8</td>
<td>CHAR</td>
<td>Error ID</td>
</tr>
<tr>
<td>ERROR_TYPE</td>
<td>49</td>
<td>1</td>
<td>CHAR</td>
<td>Error type</td>
</tr>
<tr>
<td>ERROR_CLASS</td>
<td>51</td>
<td>1</td>
<td>CHAR</td>
<td>Error class</td>
</tr>
<tr>
<td>RESOURCE_NAME</td>
<td>53</td>
<td>14</td>
<td>CHAR</td>
<td>Resource name</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>68</td>
<td>*</td>
<td>CHAR(*)</td>
<td>Error descr.</td>
</tr>
</tbody>
</table>
XPERF_CPU

This CPU performance record definition is used for selecting records with record type PERF and record subtype VM.

Main section

<table>
<thead>
<tr>
<th>Field name</th>
<th>Offset</th>
<th>Length</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RECORD_LENGTH</td>
<td>0</td>
<td>2</td>
<td>BINARY</td>
<td>Record length</td>
</tr>
<tr>
<td>NODE_NAME</td>
<td>4</td>
<td>16</td>
<td>CHAR</td>
<td>Node name</td>
</tr>
<tr>
<td>RECORD_TYPE</td>
<td>21</td>
<td>5</td>
<td>CHAR</td>
<td>Record type</td>
</tr>
<tr>
<td>DTE</td>
<td>27</td>
<td>6</td>
<td>CHAR</td>
<td>Date</td>
</tr>
<tr>
<td>TME</td>
<td>33</td>
<td>6</td>
<td>TIME(HHMMSS)</td>
<td>Time</td>
</tr>
<tr>
<td>RECORD_SUBTYPE</td>
<td>40</td>
<td>2</td>
<td>CHAR</td>
<td>Record subtype</td>
</tr>
<tr>
<td>INTERVAL</td>
<td>43</td>
<td>4</td>
<td>EXTERNAL INTEGER</td>
<td>Measured time</td>
</tr>
<tr>
<td>MEMORY_MIN</td>
<td>48</td>
<td>9</td>
<td>EXTERNAL INTEGER</td>
<td>Free pages, min</td>
</tr>
<tr>
<td>MEMORY_AVG</td>
<td>58</td>
<td>11</td>
<td>EXTERNAL FLOAT</td>
<td>Free pages, avg</td>
</tr>
<tr>
<td>MEMORY_MAX</td>
<td>70</td>
<td>9</td>
<td>EXTERNAL INTEGER</td>
<td>Free pages, max</td>
</tr>
<tr>
<td>PAGING_MIN</td>
<td>80</td>
<td>7</td>
<td>EXTERNAL INTEGER</td>
<td>Paging rate, min</td>
</tr>
<tr>
<td>PAGING_AVG</td>
<td>88</td>
<td>9</td>
<td>EXTERNAL FLOAT</td>
<td>Paging rate, avg</td>
</tr>
<tr>
<td>PAGING_MAX</td>
<td>98</td>
<td>7</td>
<td>EXTERNAL INTEGER</td>
<td>Paging rate, max</td>
</tr>
<tr>
<td>CPU_MIN</td>
<td>106</td>
<td>7</td>
<td>EXTERNAL INTEGER</td>
<td>CPU usage, min</td>
</tr>
<tr>
<td>CPU_AVG</td>
<td>114</td>
<td>9</td>
<td>EXTERNAL FLOAT</td>
<td>CPU usage, avg</td>
</tr>
<tr>
<td>CPU_MAX</td>
<td>124</td>
<td>7</td>
<td>EXTERNAL INTEGER</td>
<td>CPU usage, max</td>
</tr>
<tr>
<td>USERS</td>
<td>132</td>
<td>5</td>
<td>EXTERNAL INTEGER</td>
<td>Number of users</td>
</tr>
<tr>
<td>PROCESSES</td>
<td>138</td>
<td>5</td>
<td>EXTERNAL INTEGER</td>
<td>Number of processes</td>
</tr>
</tbody>
</table>

XPERF_DISK_FS

This disk space performance record definition is used for selecting records with record type DISK and record subtype FS.

Main section

<table>
<thead>
<tr>
<th>Field name</th>
<th>Offset</th>
<th>Length</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RECORD_LENGTH</td>
<td>0</td>
<td>2</td>
<td>BINARY</td>
<td>Record length</td>
</tr>
<tr>
<td>NODE_NAME</td>
<td>4</td>
<td>16</td>
<td>CHAR</td>
<td>Node name</td>
</tr>
<tr>
<td>RECORD_TYPE</td>
<td>21</td>
<td>5</td>
<td>CHAR</td>
<td>Record type</td>
</tr>
<tr>
<td>DTE</td>
<td>27</td>
<td>6</td>
<td>CHAR</td>
<td>Date</td>
</tr>
<tr>
<td>TME</td>
<td>33</td>
<td>6</td>
<td>TIME(HHMMSS)</td>
<td>Time</td>
</tr>
<tr>
<td>RECORD_SUBTYPE</td>
<td>40</td>
<td>2</td>
<td>CHAR</td>
<td>Record subtype</td>
</tr>
</tbody>
</table>

Record section FILE_SYSTEM

Offset 42
XPERF_DISK_FS

Length

24

<table>
<thead>
<tr>
<th>Field name</th>
<th>Offset</th>
<th>Length</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOTAL_SPACE_KB</td>
<td>1</td>
<td>9</td>
<td>EXTERNAL INTEGER</td>
<td>Total space, in kilobytes</td>
</tr>
<tr>
<td>FREE_SPACE_KB</td>
<td>11</td>
<td>9</td>
<td>EXTERNAL INTEGER</td>
<td>Free space, in kilobytes</td>
</tr>
<tr>
<td>USED_SPACE_PCT</td>
<td>21</td>
<td>3</td>
<td>EXTERNAL INTEGER</td>
<td>Used space, in percent</td>
</tr>
</tbody>
</table>

**Record section DEVICE_NAME**

Offset 66

Length

Length of DEVICE_NAME_L

<table>
<thead>
<tr>
<th>Field name</th>
<th>Offset</th>
<th>Length</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEVICE_NAME_L</td>
<td>1</td>
<td>4</td>
<td>EXTERNAL INTEGER</td>
<td>Length of section DEVICE_NAME</td>
</tr>
<tr>
<td>DEVICE_NAME</td>
<td>6</td>
<td>*</td>
<td>CHAR(+)</td>
<td>Device name</td>
</tr>
</tbody>
</table>

**Record section SPACE_NAME**

Offset 66

Length

Length of SPACE_NAME_L

<table>
<thead>
<tr>
<th>Field name</th>
<th>Offset</th>
<th>Length</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPACE_NAME_L</td>
<td>1</td>
<td>4</td>
<td>EXTERNAL INTEGER</td>
<td>Length of section SPACE_NAME</td>
</tr>
<tr>
<td>SPACE_NAME</td>
<td>6</td>
<td>*</td>
<td>CHAR(+)</td>
<td>Space name</td>
</tr>
</tbody>
</table>

**XPERF_DISK_VG**

This disk space performance record definition is used for selecting records with record type DISK and record subtype VG.

**Main section**

<table>
<thead>
<tr>
<th>Field name</th>
<th>Offset</th>
<th>Length</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RECORD_LENGTH</td>
<td>0</td>
<td>2</td>
<td>BINAR</td>
<td>Record length</td>
</tr>
<tr>
<td>NODE_NAME</td>
<td>4</td>
<td>16</td>
<td>CHAR</td>
<td>Node name</td>
</tr>
<tr>
<td>RECORD_TYPE</td>
<td>21</td>
<td>5</td>
<td>CHAR</td>
<td>Record type</td>
</tr>
<tr>
<td>DTE</td>
<td>27</td>
<td>6</td>
<td>CHAR</td>
<td>Date</td>
</tr>
<tr>
<td>TME</td>
<td>33</td>
<td>6</td>
<td>TIME(HHMMSS)</td>
<td>Time</td>
</tr>
<tr>
<td>RECORD_SUBTYPE</td>
<td>40</td>
<td>2</td>
<td>CHAR</td>
<td>Record subtype</td>
</tr>
</tbody>
</table>

**Record section VOLUME_GROUP**

Offset 42

Length

30
## XPERF_DISK_VG

<table>
<thead>
<tr>
<th>Field name</th>
<th>Offset</th>
<th>Length</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOTAL_SPACE_MB</td>
<td>1</td>
<td>9</td>
<td>EXTERNAL INTEGER</td>
<td>Total space, in megabytes</td>
</tr>
<tr>
<td>FREE_SPACE_MB</td>
<td>11</td>
<td>9</td>
<td>EXTERNAL INTEGER</td>
<td>Free space, in megabytes</td>
</tr>
<tr>
<td>USED_SPACE_MB</td>
<td>21</td>
<td>9</td>
<td>EXTERNAL INTEGER</td>
<td>Used space, in megabytes</td>
</tr>
</tbody>
</table>

## Record section SPACE_NAME

**Offset** Dependent upon length of previous section  
**Length** Length of SPACE_NAME_L

<table>
<thead>
<tr>
<th>Field name</th>
<th>Offset</th>
<th>Length</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPACE_NAME_L</td>
<td>1</td>
<td>4</td>
<td>EXTERNAL INTEGER</td>
<td>Length of section SPACE_NAME_NAME</td>
</tr>
<tr>
<td>SPACE_NAME</td>
<td>6</td>
<td>*</td>
<td>CHAR(*)</td>
<td>Space name</td>
</tr>
</tbody>
</table>

## XPERF_DISKIO

This disk I/O performance record definition is used for selecting records with record type PERF and record subtype IO.

### Main section

<table>
<thead>
<tr>
<th>Field name</th>
<th>Offset</th>
<th>Length</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RECORD_LENGTH</td>
<td>0</td>
<td>2</td>
<td>BINARY</td>
<td>Record length</td>
</tr>
<tr>
<td>NODE_NAME</td>
<td>4</td>
<td>16</td>
<td>CHAR</td>
<td>Node name</td>
</tr>
<tr>
<td>RECORD_TYPE</td>
<td>21</td>
<td>5</td>
<td>CHAR</td>
<td>Record type</td>
</tr>
<tr>
<td>DTE</td>
<td>27</td>
<td>6</td>
<td>CHAR</td>
<td>Date</td>
</tr>
<tr>
<td>TME</td>
<td>33</td>
<td>6</td>
<td>TIME(HHMMSS)</td>
<td>Time</td>
</tr>
<tr>
<td>RECORD_SUBTYPE</td>
<td>40</td>
<td>2</td>
<td>CHAR</td>
<td>Record subtype</td>
</tr>
<tr>
<td>INTERVAL</td>
<td>43</td>
<td>4</td>
<td>EXTERNAL INTEGER</td>
<td>Measurement interval</td>
</tr>
<tr>
<td>MINREAD</td>
<td>48</td>
<td>9</td>
<td>EXTERNAL INTEGER</td>
<td>KB read, minimum</td>
</tr>
<tr>
<td>AVGREAD</td>
<td>58</td>
<td>9</td>
<td>EXTERNAL FLOAT</td>
<td>KB read, average</td>
</tr>
<tr>
<td>MAXREAD</td>
<td>68</td>
<td>9</td>
<td>EXTERNAL INTEGER</td>
<td>KB read, maximum</td>
</tr>
<tr>
<td>TOTREAD</td>
<td>78</td>
<td>8</td>
<td>EXTERNAL INTEGER</td>
<td>KB read, total</td>
</tr>
<tr>
<td>MINWRITE</td>
<td>87</td>
<td>7</td>
<td>EXTERNAL INTEGER</td>
<td>KB write, minimum</td>
</tr>
<tr>
<td>AVGWRITE</td>
<td>95</td>
<td>9</td>
<td>EXTERNAL FLOAT</td>
<td>KB write, average</td>
</tr>
<tr>
<td>MAXWRITE</td>
<td>105</td>
<td>7</td>
<td>EXTERNAL INTEGER</td>
<td>KB write, maximum</td>
</tr>
<tr>
<td>TOTWRITE</td>
<td>113</td>
<td>8</td>
<td>EXTERNAL INTEGER</td>
<td>KB write, total</td>
</tr>
<tr>
<td>MINRW</td>
<td>122</td>
<td>7</td>
<td>EXTERNAL INTEGER</td>
<td>KB read and written, minimum</td>
</tr>
<tr>
<td>AVGRW</td>
<td>130</td>
<td>9</td>
<td>EXTERNAL FLOAT</td>
<td>KB read and written, average</td>
</tr>
<tr>
<td>Field name</td>
<td>Offset</td>
<td>Length</td>
<td>Format</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>--------</td>
<td>--------</td>
<td>-------------</td>
<td>--------------------------------------</td>
</tr>
<tr>
<td>MAXRW</td>
<td>140</td>
<td>7</td>
<td>EXTERNAL INTEGER</td>
<td>KB read and written, maximum</td>
</tr>
<tr>
<td>TOTRW</td>
<td>148</td>
<td>8</td>
<td>EXTERNAL INTEGER</td>
<td>KB read and written, total</td>
</tr>
<tr>
<td>BUSY</td>
<td>157</td>
<td>5</td>
<td>EXTERNAL FLOAT</td>
<td>Busy %</td>
</tr>
<tr>
<td>DISK</td>
<td>163</td>
<td>*</td>
<td>CHAR(*)</td>
<td>Physical disk name</td>
</tr>
</tbody>
</table>

**XPERF_PAGING**

This paging space performance record definition is used for selecting records with record type PERF and record subtype PS.

**Main section**

<table>
<thead>
<tr>
<th>Field name</th>
<th>Offset</th>
<th>Length</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RECORD_LENGTH</td>
<td>0</td>
<td>2</td>
<td>BINARY</td>
<td>Record length</td>
</tr>
<tr>
<td>NODE_NAME</td>
<td>4</td>
<td>16</td>
<td>CHAR</td>
<td>Node name</td>
</tr>
<tr>
<td>RECORD_TYPE</td>
<td>21</td>
<td>5</td>
<td>CHAR</td>
<td>Record type</td>
</tr>
<tr>
<td>DTE</td>
<td>27</td>
<td>6</td>
<td>CHAR</td>
<td>Date</td>
</tr>
<tr>
<td>TME</td>
<td>33</td>
<td>6</td>
<td>TIME(HHMMSS)</td>
<td>Time</td>
</tr>
<tr>
<td>RECORD_SUBTYPE</td>
<td>40</td>
<td>2</td>
<td>CHAR</td>
<td>Record subtype</td>
</tr>
<tr>
<td>PAGE_SPACE</td>
<td>43</td>
<td>20</td>
<td>CHAR</td>
<td>Page space name</td>
</tr>
<tr>
<td>DISK</td>
<td>64</td>
<td>20</td>
<td>CHAR</td>
<td>Physical disk name</td>
</tr>
<tr>
<td>PSSIZE</td>
<td>85</td>
<td>5</td>
<td>EXTERNAL INTEGER</td>
<td>Page space size in KB</td>
</tr>
<tr>
<td>PSUSE</td>
<td>91</td>
<td>5</td>
<td>EXTERNAL INTEGER</td>
<td>Page space used in KB</td>
</tr>
</tbody>
</table>

**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**

<table>
<thead>
<tr>
<th>Column</th>
<th>Position</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>0</td>
<td>2</td>
<td>DD to indicate this is a disk device record.</td>
</tr>
<tr>
<td>Date</td>
<td>3</td>
<td>8</td>
<td>YYYYMMDD of the record.</td>
</tr>
<tr>
<td>Time</td>
<td>12</td>
<td>6</td>
<td>HHMMSS of the record.</td>
</tr>
<tr>
<td>Node name</td>
<td>19</td>
<td>16</td>
<td>Node name of the machine.</td>
</tr>
<tr>
<td>Device name</td>
<td>36</td>
<td>8</td>
<td>Device name.</td>
</tr>
<tr>
<td>Total size</td>
<td>45</td>
<td>9</td>
<td>Zero-padded device size in megabytes (10242 bytes).</td>
</tr>
</tbody>
</table>
WIN_PERF_DD

<table>
<thead>
<tr>
<th>Column</th>
<th>Position</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Free space</td>
<td>55</td>
<td>9</td>
<td>Zero-padded device free space in megabytes.</td>
</tr>
<tr>
<td>Free %</td>
<td>65</td>
<td>3</td>
<td>Zero-padded percentage free.</td>
</tr>
</tbody>
</table>

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:

```csharp
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

<table>
<thead>
<tr>
<th>Column</th>
<th>Position</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>0</td>
<td>2</td>
<td>PU to indicate this is a performance usage record.</td>
</tr>
<tr>
<td>Date</td>
<td>3</td>
<td>8</td>
<td>YYYYMMDD of the record.</td>
</tr>
<tr>
<td>Time</td>
<td>12</td>
<td>6</td>
<td>HHMMSS of the record.</td>
</tr>
<tr>
<td>Node name</td>
<td>19</td>
<td>16</td>
<td>Node name of the machine.</td>
</tr>
<tr>
<td>CPU count</td>
<td>36</td>
<td>2</td>
<td>Number of CPUs in this system.</td>
</tr>
<tr>
<td>Average CPU usage</td>
<td>39</td>
<td>3</td>
<td>Zero-padded average CPU percentage usage for period. This is the average of all the snapshots in the period.</td>
</tr>
<tr>
<td>Minimum CPU usage</td>
<td>43</td>
<td>3</td>
<td>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.</td>
</tr>
<tr>
<td>Maximum CPU usage</td>
<td>47</td>
<td>3</td>
<td>Zero-padded maximum CPU percentage usage for period. This is the minimum snapshot for the period.</td>
</tr>
<tr>
<td>Memory size</td>
<td>51</td>
<td>9</td>
<td>Zero-padded total memory size at end of period, in kilobytes.</td>
</tr>
<tr>
<td>Average memory</td>
<td>61</td>
<td>9</td>
<td>Zero-padded average memory free for period, in kilobytes.</td>
</tr>
<tr>
<td>Minimum memory</td>
<td>71</td>
<td>9</td>
<td>Zero-padded minimum memory free for period, in kilobytes (1024 bytes).</td>
</tr>
<tr>
<td>Maximum memory</td>
<td>81</td>
<td>9</td>
<td>Zero-padded maximum memory free for period, in kilobytes.</td>
</tr>
</tbody>
</table>
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:

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/t/handbook/home.html](http://www-304.ibm.com/support/customercare/sas/t/handbook/home.html).

---

**Receiving weekly support updates**

To receive weekly e-mail notifications about fixes and other software support news, follow these steps:


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](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](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](https://techsupport.services.ibm.com/ssr/login).


- 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](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](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](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](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

<table>
<thead>
<tr>
<th>Title</th>
<th>SH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administration Guide and Reference, SH19-6816</td>
<td>SH19-6816</td>
</tr>
<tr>
<td>Resource Accounting for z/OS, SH19-4495</td>
<td>SH19-4495</td>
</tr>
<tr>
<td>AS/400 System Performance Feature Guide and Reference, SH19-4019</td>
<td>SH19-4019</td>
</tr>
<tr>
<td>CICS Performance Feature Guide and Reference, SH19-6820</td>
<td>SH19-6820</td>
</tr>
<tr>
<td>Distributed Systems Performance Feature Guide and Reference, SH19-4018</td>
<td>SH19-4018</td>
</tr>
<tr>
<td>Guide to Reporting, SH19-6842</td>
<td>SH19-6842</td>
</tr>
<tr>
<td>IMS Performance Feature Guide and Reference, SH19-6825</td>
<td>SH19-6825</td>
</tr>
<tr>
<td>Language Guide and Reference, SH19-6817</td>
<td>SH19-6817</td>
</tr>
<tr>
<td>Messages and Problem Determination, SH19-6902</td>
<td>SH19-6902</td>
</tr>
<tr>
<td>Network Performance Feature Installation and Administration, SH19-6901</td>
<td>SH19-6901</td>
</tr>
<tr>
<td>Network Performance Feature Reference, SH19-6822</td>
<td>SH19-6822</td>
</tr>
<tr>
<td>Network Performance Feature Reports, SH19-6821</td>
<td>SH19-6821</td>
</tr>
<tr>
<td>Resource Accounting, SH19-6818</td>
<td>SH19-6818</td>
</tr>
<tr>
<td>System Performance Feature Guide, SH19-6819</td>
<td>SH19-6819</td>
</tr>
<tr>
<td>System Performance Feature Reference Volume I, SH19-4494</td>
<td>SH19-4494</td>
</tr>
<tr>
<td>System Performance Feature Reference Volume II, SC23-7966</td>
<td>SC23-7966</td>
</tr>
</tbody>
</table>
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 feature UNIX 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 a 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.

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."

Data Language/I (DL/I)
An IBM database-management facility.

control table
A predefined Tivoli Decision Support for z/OS table that controls results returned by some log collector functions.
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.
Glossary

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

A
accessibility xv
accounting component
data tables 134, 135, 136
  reports 152, 154, 155, 157, 158, 159, 160, 161, 162
  tables 134
accounting component data tables 134, 136, 137
accounting component reports 152
accounting for AIX
  setting up 91
accounting for HP
  setting up 93
accounting for Sun Solaris
  setting up 94
accounting function in UNIX,
  implementing 91
ACCT_COMMAND 201
ACCT_CONNECT 202
ACCT_DISK 202
ACCT_PRINT 203
agents 4, 81
AIX commands to use if UNIX
  Performance component installed 102
attributes, report 54, 152
audience for this book, intended xi

B
backup and delete of generation data sets 109
books xiii
books for Administration Guide and Reference 221

C
capturing and transmitting
  Windows data
    z/OS 36
capturing and transmitting to z/OS 90
capturing UNIX data 90, 111
capturing Windows data 121
COLLECT job (JCL) 40, 107, 116, 125
collect procedure, description 46, 130
collecting
data 40
collecting performance data 4, 81
collection optimization, data 108
communication between UNIX and MVS 88, 89, 95, 100, 110, 113
communication between Windows and MVS 120
component
distributed
  introducing 3
  Linux performance
    installation 85

component (continued)
  UNIX performance
    installation 85
    introducing 81
  Windows
    installing 117
component installation
  UNIX Performance 102
CONFIG_HW 203
CONFIG_SW 203
configuration component
data tables 137
  reports 163, 164, 165, 166
  tables 137
configuration component data tables 137
configuration component reports 163
configure auto-start on the distributed	node (Solaris) 29
configure the distributed node
  (Solaris) 27
configuring
typeface xv

D
daily reports
  Distrib. CPU Util. by System, Daily Overview (DIST02) 55
  Distrib. Disk I/O for System, Daily Overview (DIST07) 61
  Distrib. File Systems, Daily Overview (DIST05) 59
  Distrib. Statistics by System, Daily Overview (DIST04) 58
  UNIX Acct Cmd Resource Consumption, Daily Overview (XACCT03) 155
  UNIX Acct Commands by User, Daily Overview (XACCT01) 152
  UNIX Acct Users and Connects, Daily Overview (XACCT07) 160
  UNIX Acct Users by Command, Daily Overview (XACCT02) 154
  UNIX Error by Class, Daily Overview (XERROR03) 169
  UNIX Error by ID, Daily Overview (XERROR01) 168
  UNIX Error by Resource, Daily Overview (XERROR04) 170
  UNIX Error by Type, Daily Overview (XERROR02) 168
  UNIX Perf CPU Utilization by System, Daily Overview (XPERF02) 174
  UNIX Perf Disk I/O for System, Daily Overview (XPERF07) 180
  UNIX Perf Statistics all Systems, Daily Overview (XPERF04) 176
  UNIX Perf Vol Group and File Syst, Daily Overview (XPERF05) 177
data
  sources of performance 81
data collection, optimizing 108
data flow and objects 45, 129
data flow from Distributed Systems
  Performance component to Tivoli Decision Support for z/OS
general flow 129
data flow from Distributed Systems
  Performance feature to Tivoli Decision Support for z/OS
general flow 45
data tables 47, 133
  in the accounting component 134, 135, 136, 137
  in the configuration component 137
  in the Distributed component 48, 49, 50
  in the error component 138
  in the performance component 138, 139, 140, 141
  in the Windows component 51, 143
  Linux 147
  LINUX component
    XCONF_HARDWARE 147
    XCONF_SOFTWARE 147
data tables (continued)  
LINUX component (continued)  
XDTPM_INFO 148  
XPERF_PS_INFO 142, 148  
XPERF_VM_INFO 142, 148  
XWTPM_INFO 149  
naming standard 47, 133  
summarization-level suffixes 47, 133  
Windows component 51, 143  
WIN PERF PU D 51, 143  
WIN PERF PU H 51, 143  
WIN PERF PU M 51, 143  
define the receiver of UNIX Performance component data 95  
definitions, log record 73, 201  
description of daily tasks performed by UNIX Performance component 99  
description of UNIX Performance component files 101  
disk blocks  
allocated to node 159  
allocated to user 158  
disk device record  
WIN PERF DD 76, 209  
disk space  
for installation 87, 119  
utilization 60, 61, 178, 180  
DIST_DISK_FS_A 74  
DIST PERF IO A 74  
DIST PERF PS A 75  
DIST PERF VM A 73  
Distrib. CPU Util. for System, Hourly Trend  
reports in the Distributed component 54  
Distributed component configuring 7  
data tables 48, 49  
DIST CPU M 48  
installing 7  
planning 7  
reports 54, 55, 56, 58, 59, 60, 61, 62, 63  
tables 48  
Distributed component data tables 48, 50  
Distributed component reports 54  
Distributed Systems Performance feature introducing 3, 81  
documentation, TDS 221  
drlpsperfd 102  

E  
education xv  
elapsed time 132, 154, 155, 157  
ERROR 205  
error component data tables 138  
reports 167, 168, 169, 170, 171  
tables 138  
error component data tables 138  
error component reports 167  
errors (continued)  
by ID  
number 168  
by resource  
number 170  
by type  
number 168  
errors, by resource  
monthly trend 171  
evaluation and planning 86, 119  
exception reports  
Distrib. CPU Util. Exceptions (DIST09) 63  
F  
file transfer 95, 112  
fixes, obtaining 213  
format, Tivoli Decision Support for z/OS report 53, 151  
G  
gather of data 4, 81  
Generation Data Group (GDG) 106  
glossary 223  
groups, report 53, 152  
H  
hardware configuration  
for a device class 164  
for a system 163  
hourly reports  
Distrib. Disk I/O for Disk, Hourly Trend (DIST06) 60  
Distrib. Page Space Util., Hourly Trend (DIST08) 62  
Distrib. Statistics for System, Hourly Trend (DIST03) 56  
UNIX Perf CPU Utilization by System, Hourly Trend (XPERF01) 173  
UNIX Perf Disk I/O for a Disk, Hourly Trend (XPERF06) 178  
UNIX Perf Page Space Utilization, Hourly Trend (XPERF08) 181  
UNIX Perf Statistics by System, Hourly Trend (XPERF03) 175  
how to use this book xi  
I  
I/O (kilobytes) 152, 154, 155, 157, 178, 180  
identifiers, report 53, 151  
implementation process planning  
Distributed component 7  
implementation process, planning 85, 117  
information centers, searching for problem resolution 213  
information logs 101  
installation requirements 87, 110, 119  
installation (continued)  
AIX 21  
HP-UX 17  
making input data available 106  
optimizing data collection 108  
putting component into production 41  
putting feature into production 109, 126  
RHEL 12  
SLES 8  
Solaris 25  
updating lookup tables 104  
Windows 35  
testing 31  
testing the installation 41, 109, 125  
UNIX performance component 85  
installing  
Distributed component 7  
AIX 21  
distributed 7  
HP-UX 16  
RHEL 12  
SLES 8  
Solaris 25  
z/OS 29  
planning subcomponents  
Windows 34  
Tivoli Decision Support for z/OS  
AIX 22  
HP-UX 17  
RHEL 13  
SLES 9  
Solaris 26  
Windows component 33  
Windows 35  
z/OS 38  
installing the Linux Performance component on your Linux nodes 110  
installing the UNIX Performance component on your UNIX nodes 87  
installing the Windows component on your Windows nodes 119  
installing UNIX Performance component on the z/OS system 102  
installing Windows component on the z/OS system 123  
intended audience for this book xi  
Internet searching for problem resolution 213  
invoke drlpsperfd at start time 99  
J  
JCL  
COLLECT with specified Tivoli Decision Support for z/OS tables 108  
General COLLECT 40, 107, 116, 125  
K  
knowledge bases, searching for problem resolution 213
L

data tables 147
subcomponents to install 110
Linux component
commands 115
transmitting data to z/OS 111
LINUX component
data tables
XCONF_HARDWARE 147
XCONF_SOFTWARE 147
XDISK_INFOR 147
XDTEMP_INFOR 148
XPERF_PS_INFO 142, 148
XPERF_VM_INFO 142, 148
XWTMP_INFOR 149
report IDs
XLINU01 182
XLINU02 183
XLINU03 184
XLINU04 185
XLINU05 186
XLINU06 187
XLINU07 188
XLINU08 189
XLINU09 190
XLINU10 191
reports
Disk Space Allocation 183
Hardware Configuration 186
Percentile Work Size 182
Performance 'PS' for Volumes
Info 185
Performance 'VM' for CPU 191
Performance 'VM' for Swap
Memory 190
Performance from User
Memory 184
Process Information 189
Software Configuration 187
User Information 188
Linux performance component
installation 85
Linux Performance component
customizing parameters 113
list of terms used in this book 223
log definitions, Tivoli Decision Support
for z/OS 131
log record definitions 73, 201
logs
UNIX 131
lookup tables 104, 133

M

manuals xiii
manuals, TDS 221
monthly reports
UNIX Acct Disk Blocks by User,
Monthly Overview (XACCT05) 158
UNIX Acct Disk Blocks in 1000s,
Monthly Trend (XACCT06) 159
UNIX Acct Printed Pages by System,
Monthly Overview (XACCT09) 162
UNIX Acct Printed Pages by User,
Monthly Overview (XACCT08) 161

cmply reports (continued)
UNIX Acct User Resource Usage,
 Monthly Overview (XACCT04) 157
UNIX Error by Resource, Monthly
Trend (XERROR05) 171

N

naming standards for tables 47, 133

O

objects and data flow 45, 129
online publications
accessing xiv
operating the UNIX Performance
component on UNIX 100
optimizing data collection 108
overview reports
Distrib. CPU Util. by System, Daily
Overview (DIST02) 55
Distrib. Disk I/O for System, Daily
Overview (DIST07) 61
Distrib. File Systems, Daily Overview
(DIST05) 59
Distrib. Statistics by System, Daily
Overview (DIST04) 58
UNIX Acct Cmd Resource
Consumption, Daily Overview
(XACCT03) 155
UNIX Acct Commands by User, Daily
Overview (XACCT01) 152
UNIX Acct Disk Blocks by User,
Monthly Overview (XACCT05) 158
UNIX Acct Printed Pages by System,
Monthly Overview (XACCT09) 162
UNIX Acct Printed Pages by User,
Monthly Overview (XACCT08) 161
UNIX Acct User Resource Usage,
Monthly Overview (XACCT04) 157
UNIX Acct Users and Connects, Daily
Overview (XACCT07) 160
UNIX Acct Users by Command, Daily
Overview (XACCT02) 154
UNIX Configuration of HW for a
System, Overview
(XCONFIG01) 163
UNIX Configuration of HW for
Device Class, Overview
(XCONFIG02) 164
UNIX Configuration of SW for a
System, Overview
(XCONFIG03) 165
UNIX Configuration of SW for Object,
Overview (XCONFIG04) 166
UNIX Error by Class, Daily Overview
(XERROR03) 169
UNIX Error by ID, Daily Overview
(XERROR01) 168
UNIX Error by Resource, Daily
Overview (XERROR04) 170
UNIX Error by Type, Daily Overview
(XERROR02) 168
UNIX Perf CPU Utilization by System,
Daily Overview (XPERF02) 174

P

page space, utilization 56, 58, 62, 175,
176, 181
paging rate 56, 58, 175, 176
parameter file values for all
subcomponents 96
parameter file values for Performance
component 97
parameter files for all subcomponents
Linux Performance component 113
PERF_CPU 206
PERF_DISK_FS 206
PERF_DISK_IO 208
PERF_DISK_VG 207
PERF_PAGING 209
performance component
data tables 139, 140
reports 172, 173, 174, 175, 176, 177,
178, 180, 181
tables 138
XPERF_CPU_M 139
Performance component
parameter file values 97
performance component data tables
138, 141
performance component reports 172
performance management
collecting data (overview) 4, 81
performance usage record
WIN_PERF PU 76, 210
planning
implementation process
Distributed component 7
planning and evaluation 86, 119
planning the implementation process
33, 85, 117
print resources used
pages, by user 161, 162
printouts, by user 162
problem determination
describing problems 216
determining business impact 215
submitting problems 216
processes, number of 56, 58, 175, 176
production component, putting into
32
publications xiii
TDS 221

R

record definition descriptions
ACCT_COMMAND 201
ACCT_CONNECT 202
ACCT_DISK 202
ACCT_PRINT 203
CONFIG_HW 203

Index 229
record definition descriptions (continued)

CONFIG_SW 203
DIST_DISK_FS_A 74
DIST_PERF_IO_A 74
DIST_PERF_FS_A 75
DIST_PERF_VM_A 73
ERROR 205
PERF_CPU 206
PERF_DISK_FS 206
PERF_DISK_IO 208
PERF_DISK_VG 207
PERF_PAGING 209

record definitions for Tivoli Decision Support for z/OS

Support for z/OS 131

report format 53, 151
groups 5, 82

report IDs 53, 151

DIST02 (Distrib. CPU Util. by System, Daily Overview) 55
DIST03 (Distrib. Statistics for System, Hourly Trend) 56
DIST04 (Distrib. Statistics by System, Daily Overview) 58
DIST05 (Distrib. File Systems, Daily Overview) 59
DIST06 (Distrib. Disk I/O for Disk, Hourly Trend) 60
DIST07 (Distrib. Disk I/O for System, Daily Overview) 61
DIST08 (Distrib. Page Space Util., Hourly Trend) 62
DIST09 (Distrib. CPU Util. Exceptions) 63
XACCT01 (UNIX Acct Commands by User, Daily Overview) 152
XACCT02 (UNIX Acct Users by Command, Daily Overview) 154
XACCT03 (UNIX Acct Cmd Resource Consumption, Daily Overview) 155
XACCT04 (UNIX Acct User Resource Usage, Monthly Overview) 157
XACCT05 (UNIX Acct Disk Blocks by User, Monthly Overview) 158
XACCT06 (UNIX Acct Disk Blocks in 100s, Monthly Trend) 159
XACCT07 (UNIX Acct Users and Connects, Daily Overview) 160
XACCT08 (UNIX Acct Printed Pages by User, Monthly Overview) 161
XACCT09 (UNIX Acct Printed Pages by System, Monthly Overview) 162
XCONFIG01 (UNIX Configuration of HW for a System, Overview) 163
XCONFIG02 (UNIX Configuration of HW for Device Class, Overview) 164
XCONFIG03 (UNIX Configuration of SW for a System, Overview) 165
XCONFIG04 (UNIX Configuration of SW for Object, Overview) 166
XERROR01 (UNIX Error by ID, Daily Overview) 168
XERROR02 (UNIX Error by Type, Daily Overview) 168
XERROR03 (UNIX Error by Class, Daily Overview) 169

report IDs (continued)

XERROR04 (UNIX Error by Resource, Daily Overview) 170
XERROR05 (UNIX Error by Resource, Monthly Trend) 171
XPERF01 (UNIX Perf CPU Utilization by System, Hourly Trend) 173
XPERF02 (UNIX Perf CPU Utilization by System, Daily Overview) 174
XPERF03 (UNIX Perf Statistics by System, Hourly Trend) 175
XPERF04 (UNIX Perf Statistics all Systems, Daily Overview) 176
XPERF05 (UNIX Perf Vol Group and File Syst, Daily Overview) 177
XPERF06 (UNIX Perf Disk I/O for a Disk, Hourly Trend) 178
XPERF07 (UNIX Perf Disk I/O for System, Daily Overview) 180
XPERF08 (UNIX Perf Page Space Utilization, Hourly Trend) 181

reports 53, 151

attributes 54, 152

format and general description 53, 151

identifiers, unique 53, 151

in the accounting component 152, 154, 155, 157, 158, 159, 160, 161, 162

in the configuration component 163, 164, 165, 166

in the Distributed component 55

in the Distributed component 54, 56, 58, 59, 60, 61, 62, 63

in the error component 167, 168, 169, 170, 171

in the performance component 53, 152, 172, 173, 174, 175, 176, 177, 178, 180, 181

in the Windows component 64, 192

Windows CPU Utilization by System, Daily Overview 67, 195

Windows CPU Utilization by System, Hourly Trend 66

Windows Disk Usage for Device, Daily Overview 65, 193

Windows Disk Usage for System, Hourly Trend 65, 192

Windows Memory Usage by System, Daily Overview 69, 197

Windows Memory Usage by System, Hourly Trend 68, 196

Windows System Overview Report 70, 198

LINUX component

Disk Space Allocation 183
Hardware Configuration 186
Percentile Work Size 182
Performance 'PS' for Volumes
Info 185
Performance 'VM' for CPU 191
Performance 'VM' for Swap Memory 190
Performance from User Memory 184
Process Information 189
Software Configuration 187
User Information 188

reports (continued)

source tables 54, 152
variables 54, 152

reports in the Distributed component

Distrib. CPU Util. for System, Hourly Trend 54

reports in the Windows component

Windows CPU Utilization by System, Hourly Trend 194

requirements

disk space

AIX 21
HP-UX 17
RHEL 13
SLES 8
Solaris 25
Windows 35

software

AIX 21
HP-UX 17
RHEL 13
SLES 9
Solaris 26
Windows 35

requirements, installation

disk space 87, 110, 119
distributed

AIX 21
HP-UX 17
RHEL 13
SLES 8
Solaris 25
software 87, 110, 120
UNIX 87, 110
Windows 119

Windows 35

retry interval, file transfer 97, 114

retry limit, file transfer 96, 114

S

Sart and check the distributed node

(Solaris) 28

set up UNIX accounting 91

setting up accounting for AIX 91

setting up accounting for HP 93

setting up accounting for Sun Solaris 94

setting up scheduled tasks 37

software configuration

for a system 165

for an object 166

Software Support

contacting 214

describing problems 216
determining business impact 215

receiving weekly updates 214

submitting problems 216

source tables, report 54, 152

space utilization 59, 177

start and check

distributed node

AIX 24
HP-UX 20

start the drperfd daemon 98

starting

distributed node

RHEL 15
UNIX lookup tables  
UNIX accounting  
subcomponents to install, LINUX  
suffixes, data-table  47, 133  
system time  152, 154, 155, 157

T

table descriptions  134  
tables  47, 48, 49, 50, 104, 105, 134, 135, 136, 137, 138, 139, 140, 141, 144, 145  
accounting component  134  
data, lookup, control  133  
in the configuration component  137  
in the Distributed component  48  
in the error component  138  
in the performance component  138  
in the Windows component  51, 143  
lookup  144  
TCP/IP  5, 82  
terms used in this book  223  
testing  
installation  31  
testing the installation  41, 109, 125  
Tivoli technical training  xv  
training, Tivoli technical  xv  
transfer of files to MVS  90, 111, 121  
transferring  
feature code to Windows  35  
trend reports  
Distrib. Disk I/O for Disk, Hourly Trend (DIST06)  60  
Distrib. Page Space Util., Hourly Trend (DIST08)  62  
Distrib. Statistics for System, Hourly Trend (DIST03)  56  
UNIX Acct Disk Blocks in 1000s, Monthly Trend (XACCT06)  159  
UNIX Error by Resource, Monthly Trend (XERROR05)  171  
UNIX Perf CPU Utilization by System, Hourly Trend (XPERF01)  173  
UNIX Perf Disk I/O for a Disk, Hourly Trend (XPERF06)  178  
UNIX Perf Page Space Utilization, Hourly Trend (XPERF08)  181  
UNIX Perf Statistics by System, Hourly Trend (XPERF03)  175  
typeface conventions  xv

U

UNIX  
operating Performance component  100  
UNIX accounting  
setting up  91  
UNIX lookup tables  105, 144, 145  
UNIX performance component  
installation  85  
UNIX Performance component  
AIX commands to use  102  
description of component files  101  
description of daily tasks performed by  99  
UNIX Performance component  
(continued)  
installing you your UNIX nodes  87  
UNIX Performance component parameters  
customizing  96  
updating lookup tables  
for UNIX component  104  
use of this book, intended xi  
user time  152, 154, 155, 157  
users, number of  56  

V

variables, report  54, 152  
view table for accounting component  137

W

WIN_PERF_DD  76, 209  
WIN_PERF PU  76, 210  
Windows  
capturing and transmitting to z/OS  121  
parameter file values  37  
Windows component  
configuring  33  
data tables  
WIN_PERF DD D  51  
WIN_PERF DD H  51, 143  
WIN_PERF DD M  51  
WIN_PERF PU D  51, 143  
WIN_PERF PU H  51, 143  
WIN_PERF PU M  51, 143  
installs  33  
planning the implementation process  33  
reports  64, 192  
Windows CPU Utilization by System, Daily Overview  67, 195  
Windows CPU Utilization by System, Hourly Trend  66  
Windows Disk Usage for Device, Daily Overview  65, 193  
Windows Disk Usage for System, Hourly Trend  65, 192  
Windows Memory Usage by System, Daily Overview  69, 197  
Windows System Overview Report  70, 198  
tables  51, 143  
Windows component data tables  51, 143  
Windows component reports  64, 192  
Windows CPU Utilization by System, Daily Overview  67, 195  
Windows CPU Utilization by System, Hourly Trend  66  
Windows Disk Usage for Device, Daily Overview  65, 193  
Windows Disk Usage for System, Hourly Trend  65, 192  
Windows Memory Usage by System, Daily Overview  69, 197

X

X_NODE_NAME lookup table  105  
XACCT_COMMAND  201  
XACCT_CONNECT  202  
XACCT_CONNECT_TYPE lookup table  105  
XACCT_DISK  202  
XACCT_PRINT  203  
XACCT_PRINT_TYPE lookup table  105  
XCONF_HARDWARE  147  
XCONF_SOFTWARE  147  
XCONFIG_HW  203  
XCONFIG_SW  203  
XDISK_INFOR  147  
XDTMP_INFOR  148  
XERROR  205  
XPERF_CPU  206  
XPERF_DISK_FS  206  
XPERF_DISK_IO  208  
XPERF_DISK_VG  207  
XPERF_PAGING  209  
XPERF_PS_INFO  148  
XPERF_VM_INFO  148  
XWTMP_INFOR  149

Z

z/OS system  
installing UNIX Performance component  102