



**IBM Tivoli Monitoring for Transaction
Performance, Version 5.3
Warehouse Enablement Pack, Version 5.3.0
Implementation Guide**

for Tivoli Data Warehouse, Version 1.2

Note:

Before using this information and the product it supports, read the information in Notices on page 137.

First Edition (September 2004)

This edition applies to IBM Tivoli Monitoring for Transaction Performance, Version 5 release 3 and to all subsequent releases and modifications until otherwise indicated in new editions.

© Copyright International Business Machines Corporation 2004. All rights reserved.

US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

1 About this guide	9
1.1 What's new in this warehouse pack	9
1.2 Who should read this guide	10
1.3 Publications	10
1.3.1 IBM Tivoli Monitoring for Transaction Performance library.....	11
1.3.2 Tivoli Data Warehouse library	11
1.3.3 Related publications.....	12
1.3.3.1 IBM Redbooks.....	12
1.3.3.2 IBM DB2 Universal Database, DB2 Data Warehouse Center, and DB2 Warehouse Manager library ...	12
1.3.4 Accessing publications online	13
1.3.5 Ordering publications	13
1.4 Accessibility.....	14
1.5 Contacting software support	14
1.6 Participating in newsgroups	14
1.7 Typeface conventions	14
 2 Overview	 16
2.1 Overview of Tivoli Data Warehouse	16
2.2 Overview of IBM Tivoli Monitoring for Transaction Performance Version 5.3 warehouse pack	17
 3 Reports.....	 21
3.1 Sample report layouts.....	29
3.1.1 Average Response Time by Application	29
3.1.2 Average Response Time by User for Transaction <transaction>	30
3.1.3 Average Response Time by Policy	31
3.1.4 Average Response Time by Policy Group.....	32
3.1.5 Average Response Time by Policy for Policy Group <i>PolicyGroupName</i>	34
3.1.6 Average Response Time by Agent Group for Transaction <transaction>	35
3.1.7 Execution Load by Agent Group For Transaction <transaction>	36
3.1.8 Execution Load by Application	38
3.1.9 Execution Load by User For Transaction <transaction>	40
3.1.10 Execution Load For Policy <i>Reservations</i>	41
3.1.11 Execution Load by Agent for Policy <i>Policy</i>	42
3.1.12 Transaction Availability For <transaction>	43
3.1.13 Overall Transaction Response Time For Transaction <transaction>	44
3.1.14 Slowest Transactions	45
3.1.15 Policy Load per Agent.....	45
3.1.16 Least Throughput.....	46
3.1.17 Enhanced <application> Availability	47
3.1.18 Resource Usage and Transaction Response Time for Transaction <i>Petstore</i>	49
3.2 Report Parameter Inputs Defined.....	50
3.2.1 Average Response Time by Application	51
3.2.1.1 Start Date and End Date.....	51
3.2.1.2 Time Granularity	51
3.2.1.3 Application Name	51
3.2.2 Average Response Time by User for Transaction <transaction>	51
3.2.2.1 Start Date and End Date.....	51
3.2.2.2 Time Granularity	51
3.2.2.3 User Name	52
3.2.2.4 Transaction Name	52
3.2.3 Average Response Time by Policy	52

3.2.3.1 Start Date and End Date.....	52
3.2.3.2 Time Granularity	52
3.2.3.3 Policy Name	52
3.2.4 Average Response Time by Policy Group.....	52
3.2.4.1 Start Date and End Date.....	52
3.2.4.2 Time Granularity	53
3.2.4.3 Policy Group Name	53
3.2.5 Average Response Time by Policy For Policy Group <policy_group>.....	53
3.2.5.1 Start Date and End Date.....	53
3.2.5.2 Time Granularity	53
3.2.5.3 Policy Group Name	53
3.2.6 Average Response Time by Agent Group for Transaction <transaction>	53
3.2.6.1 Start Date and End Date.....	54
3.2.6.2 Time Granularity	54
3.2.6.3 Agent Group Name.....	54
3.2.6.4 Transaction Name.....	54
3.2.7 Execution Load by Agent Group For Transaction <transaction>	54
3.2.7.1 Start Date and End Date.....	54
3.2.7.2 Time Granularity	54
3.2.7.3 Agent Group Name.....	55
3.2.7.4 Transaction Name.....	55
3.2.8 Execution Load by Application	55
3.2.8.1 Start Date and End Date.....	55
3.2.8.2 Time Granularity	55
3.2.8.3 Application Name.....	55
3.2.9 Execution Load by User For Transaction <transaction>	55
3.2.9.1 Start Date and End Date.....	56
3.2.9.2 Time Granularity	56
3.2.9.3 User Name	56
3.2.9.4 Transaction Name.....	56
3.2.10 Execution Load For Policy <policy>	56
3.2.10.1 Start Date and End Date.....	56
3.2.10.2 Time Granularity	56
3.2.10.3 Policy Name.....	56
3.2.11 Execution Load by Agent For Policy <policy>	57
3.2.11.1 Start Date and End Date.....	57
3.2.11.2 Time Granularity	57
3.2.11.3 Policy Name.....	57
3.2.12 Transaction Availability For Transaction <transaction>	57
3.2.12.1 Start Date and End Date.....	57
3.2.12.2 Time Granularity	58
3.2.12.3 Agent Name.....	58
3.2.12.4 Transaction Name.....	58
3.2.13 Overall Transaction Response Time For Transaction <transaction>	58
3.2.13.1 Start Date and End Date.....	58
3.2.13.2 Time Granularity	58
3.2.13.3 Agent Name	58
3.2.13.4 Transaction Name.....	59
3.2.14 Policy Load per Agent.....	59
3.2.14.1 Start Date and End Date.....	59
3.2.14.2 Time Granularity	59
3.2.14.3 Agent Name	59
3.2.15 Least Throughput.....	59
3.2.15.1 Start Date and End Date.....	59
3.2.15.2 Time Granularity	60
3.2.15.3 Agent Name	60

3.2.15.4 Application Name	60
3.2.15.5 User Name	60
3.2.15.6 Transaction Name	60
3.2.15.7 Number of Transactions.....	60
3.2.16 Slowest Transactions	60
3.2.16.1 Start Date and End Date.....	61
3.2.16.2 Time Granularity	61
3.2.16.3 Agent Name	61
3.2.16.4 Application Name	61
3.2.16.5 User Name	61
3.2.16.6 Transaction Name	61
3.2.16.7 Number of Transactions.....	61
3.2.17 Enhanced Availability	61
3.2.17.1 Start Date and End Date.....	62
3.2.17.2 Time Granularity	62
3.2.17.3 Application Name	62
3.2.18 Resource Usage, Response Time and Execution Load For Transaction <transaction>	62
3.2.18.1 Start Date and End Date.....	63
3.2.18.2 Time Granularity	63
3.2.18.3 Agent Name	63
3.2.18.4 Transaction Name	64
3.3 Report Notes and Limitations.....	64
3.3.1 Viewing reports	64
3.3.2 Report fonts	65
3.3.3 Limitations	65
4 Installing and configuring the warehouse pack	66
4.1 Prerequisite hardware and software.....	66
4.1.1 Installing the Crystal Enterprise 9 patch for ADAPT00319691	66
4.2 Product notes and limitations	67
4.2.1 Product notes	67
4.2.2 Limitations.....	67
4.2.2.1 No support for DB2 Universal Database for z/OS and OS/390.....	67
4.2.2.2 Synthetic Edge Transactions in the warehouse	67
4.2.2.3 Changing the ETL source input database	67
4.2.2.4 Rollup of Error_count data from the hourly fact table in the MART database.....	67
4.2.2.5 Reinstalling the Tivoli Monitoring for Transaction Performance warehouse pack	68
4.2.2.6 How to handle multiple failed unattended executions of the ETL.....	68
4.3 Database-sizing considerations.....	69
4.4 Pre-installation procedures.....	70
4.4.1 Tivoli Data Warehouse 1.2 required.....	70
4.4.2 Make warehouse database backups	70
4.4.3 Upgrading to the new Web Transaction Performance 5.3.0.0 warehouse pack.....	70
4.4.4 ODBC database client.....	70
4.4.5 ODBC drivers	71
4.5 Installation of the warehouse pack.....	71
4.6 Post-installation procedures	71
4.6.1 One time process to run to upgrade warehouse data.....	71
4.6.2 Scheduling warehouse pack processes.....	71
4.6.3 Change Prune Control values from installed defaults.....	72
4.6.4 Accessing the application source database	72
4.6.5 Configure warehouse source and target settings.....	73
4.7 Migration from a previous release of the warehouse pack	74
4.8 Uninstalling the warehouse pack.....	74
4.9 Multiple data centers.....	75

4.10 Multiple customer environments.....	76
5 Maintenance and problem determination	77
5.1 Backing up and restoring.....	77
5.2 Pruning data.....	77
5.2.1 Central data warehouse.....	77
5.2.1.1 Pruning measurement data (table Prune_Msmt_Control).....	77
5.2.2 Data mart	78
5.2.2.1 Pruning data mart data (table Prune_Mart_Control).....	78
5.3 Extraction control (table Extract_Control).....	79
5.3.1 Measurement Data.....	79
5.3.2 Data Mart Data	80
5.4 Maintenance scripts.....	80
5.4.1 Show_ExtCtl_Values.sql.....	81
5.4.2 Show_ExtLog_Values.sql	81
5.4.3 Show_Prune_Mart_Values.sql	81
5.4.4 Show_Prune_Msmt_Values.sql.....	81
5.4.5 Reset_ETL1_extract_window.sql.....	81
5.4.6 Reset_ETL2_extract_window.sql.....	82
5.4.7 Reset_ETL1_ETL2_extract_windows.sql.....	82
5.4.8 Reset_ETL1_ETL2_prune_values_to_clear_data.sql.....	82
5.4.9 Reset_ETL1_ETL2_prune_values_to_defaults.sql	83
5.4.10 Create_Cross_App_report_views.sql.....	83
5.4.11 Clean_and_Reset_BWM_warehouse_Delete_tables.sql	83
5.4.12 Clean_and_Reset_BWM_warehouse_Create_tables.sql	84
5.5 Problem determination	85
5.5.1 Crystal error for report Resource Usage, Response Time and Execution Load For Transaction <transaction>.....	85
5.5.2 Sample ETL Log Informational messages.....	85
6 ETL processes.....	87
6.1 BWM_c05_Upgrade_Process.....	87
6.2 BWM_c10_src_to_cdw_Process.....	91
6.3 BWM_m05_cdw_to_mart_Process	92
7 Central data warehouse information	93
7.1 Component configuration	93
7.1.1 Component type (table CompTyp)	93
7.1.2 Component extension (table Comp_ext)	94
7.1.3 Component (table Comp)	95
7.1.4 Component relationship type (table RelnTyp).....	99
7.1.5 Component relationship rule (table RelnRul).....	100
* For WebLogic, which does not have a J2EE_NODE value it has a relationship of J2EE_DOMAIN to J2EE_SERVER that the other J2EE applications cannot have.....	101
7.1.6 Component relationship (table CompReln)	101
7.1.7 Component type keyword (table CompTyp_Keyword).....	104
7.1.8 Attribute type (table AttrTyp).....	104
7.1.9 Attribute rule (table AttrRul)	105
7.1.10 Attribute domain (table AttrDom)	106
7.1.11 Component attribute (table CompAttr).....	107
7.1.12 Component type relationship (table CTypReln)	109
7.1.13 Component attribute type relationship (table ATypReln).....	109
7.2 Component measurement	109
7.2.1 Measurement group type (table MGrpTyp).....	109

7.2.2 Measurement group (table MGrp)	110
7.2.3 Measurement group member (table MGrpMbr)	110
7.2.4 Measurement unit category (table MUnitCat)	111
7.2.5 Measurement unit (table MUnit)	111
7.2.6 Measurement alias names (table MTypReIn)	112
7.2.7 Time summary (table TmSum)	112
7.2.8 Measurement source (table MSrc)	112
7.2.9 Measurement source history (table MSrcHistory)	113
7.2.10 Measurement type (table MsmtTyp)	113
7.2.11 Component measurement rule (table MsmtRul)	114
7.2.12 Measurement (table Msmt)	115
7.2.13 Threshold measurement objective (table Mobj)	116
7.2.14 Threshold measurement objective range (table MobjRng)	117
7.2.15 Threshold severity level (table SevLvl)	117
7.3 Helper tables	117
7.3.1 Component long (table BWM.COMP_ATTR_LONG)	117
7.4 Exception tables	118
7.5 Incremental extraction	119
 8 Data mart schema information	 121
8.1 Data mart BWM Transaction Performance	121
8.2 Star schemas	121
8.2.1 BWM hourly transaction performance transaction node star schema	121
8.2.1.1 Fact table BWM.F_TX_ND_HOUR	122
8.2.2 BWM daily transaction performance transaction node star schema	123
8.2.2.1 Fact table BWM.F_TX_ND_DAY	123
8.2.3 BWM weekly transaction performance transaction node star schema	124
8.2.3.1 Fact table BWM.F_TX_ND_WEEK	124
8.2.4 BWM monthly transaction performance transaction node star schema	125
8.2.4.1 Fact table BWM.F_TX_ND_MONTH	126
8.2.5 AMY IBM Tivoli Monitoring for Operating Systems star schema for cross application reporting	126
8.3 Fact staging tables	127
8.3.1 Fact staging table BWM.STG_TX_ND_HR	128
8.4 Metric dimension tables	128
8.4.1 BWM.D_TX_ND_METRIC	128
8.4.2 AMY.D_CPU_METRIC	130
8.5 Dimension tables	131
8.5.1 Dimension table BWM.D_HOST	132
8.5.2 Dimension table BWM.D_HOST_AGENT_GRPS	132
8.5.3 Dimension table BWM.D_TX	132
8.5.4 Dimension table BWM.D_TX_ND	132
8.5.5 Dimension table BWM.D_TX_ND_POL_GRPS	133
8.5.6 Dimension table BWM.D_APP	133
8.5.7 AMY.D_HOST_IP	133
8.6 Data translation tables	134
8.6.1 BWM.T_TX_ND_METRIC	134
8.6.2 BWM.T_HOST	135
8.6.3 BWM.T_HOST_AGENT_GRPS	135
8.6.4 BWM.T_TX	135
8.6.5 BWM.T_TX_ND	136
8.6.6 BWM.T_TX_ND_POL_GRPS	136
8.6.7 BWM.T_APP	136
8.7 Exception tables	137

1 About this guide

This document describes the warehouse enablement pack, Version 5.3.0 for IBM® Tivoli Monitoring for Transaction Performance™ Version 5.3. This warehouse enablement pack is created for Tivoli Data Warehouse, Version 1.2.

This warehouse enablement pack (hereafter referred to as warehouse pack) uses the product code of BWM which is used as part of the name for the scripts and processes for this warehouse pack as well as to define the schema owner of any tables created in the central data warehouse or data mart databases. This document covers the following topics:

- Installing and configuring the warehouse pack
- The data flow and data structures used by the warehouse pack
- Running the warehouse pack processes
- Creating reports on transaction performance

With this warehouse pack you can enable a set of extract, transform and load (ETL) utilities to extract and move data from an IBM Tivoli Monitoring for Transaction Performance database to the central data warehouse database. Database administrators and developers require the type of information provided in this document to install the ETL tools and to understand what data is moved into the central data warehouse by the central data warehouse ETL. The data is then used to populate a data mart database used for reporting on transaction performance.

1.1 What's new in this warehouse pack

1. This version of the warehouse pack only runs on Tivoli Data Warehouse Version 1.2 and is installed as an upgrade on an already-installed version Web Transaction Performance 5.2 of the IBM Tivoli Monitoring for Transaction Performance warehouse pack, or as a new install.
2. Web Transaction Performance 5.3 reports have been enhanced for the Web Transaction Performance 5.3 version of this warehouse pack and several new reports have been added, including a cross-application report for additional reporting capability. All the reports run with Crystal Enterprise Professional v9 for Tivoli. The cross application report uses Tivoli Monitoring for Transaction Performance and IBM Tivoli Monitoring data.
3. Although you can install Tivoli Data Warehouse, Version 1.2 on z/OS computers, this warehouse pack only runs on UNIX and Windows.
4. Additional central data warehouse tables contain the warehouse pack's product code, 'BWM' for this warehouse pack, allowing better determination of which warehouse pack data belongs to.
5. Data shared by various warehouse packs is now marked with a MSrc_Corr_Cd of 'SHARED'. The details of what may be marked as SHARED are shown below.
 - In warehouse pack Web Transaction Performance 5.3 components whose component type have an MSrc_Cd of 'MODEL1' are marked as SHARED for their MSrc_Corr_Cd column to show this component can be shared in the warehouse by any warehouse packs.
 - In warehouse pack Web Transaction Performance 5.3, the CompReln CDW table will have a MSrc_Corr_Cd of SHARED when both the source component's CompTyp is MODEL1.
6. An Exception_Log table has been added to the ETL processes to allow exception details to be written to the ETL step's output logs.
7. The first time the ETL process is run it only retrieves the data that is greater than the prune measurement value set for the central data warehouse (CDW). This prevents excessive amounts data from being extracted on the first run and causing database performance problems.

8. Staging tables and process and step names have been renamed to conform to a new warehouse naming standard for uniformity across warehouse packs.
9. New warehouse component types and relationships are defined:
 - New component types have been defined. They are BWM_AGENT_GROUP, BWM_MGMT_POLICY, and BWM_MGMT_POL_GRP. In prior versions of this warehouse pack BWM_MGMT_POLICY was an attribute of component type BWM_TX_NODE. In this version, Web Transaction Performance 5.3, BWM_MGMT_POLICY will still be an attribute of component type BWM_TX_NODE as well as a component type of its own.

Note: The AttrType of BWM_MGMT_POLICY will be deprecated in future releases of this warehouse pack, but was retained in this version for backward compatibility.

- J2EE_SERVER has a new RUNSON relationship to the host it ran on.
- ARM data is now supported as an application type for reporting.
- Fields containing IP addresses have been increased to 39 characters to support the new IP V6 format.
- Transaction abort count data has been added to the warehouse measurement data and metrics.
- A sample customer view for each star schema table, except the metric table, is provided so that customer data can be split into separate star schemas for each customer. The metrics associated with each view are the same so there is not a separate view for the metric dimension table.

1.2 Who should read this guide

This guide is for people who do any of the following activities:

- Plan for and install the warehouse pack
- Use and maintain the warehouse pack and its reports
- Create new reports
- Create additional warehouse packs that use data from this warehouse pack

Administrators and installers should have the following knowledge or experience:

- Basic system administration and file management of the operating systems on which the components of Tivoli Data Warehouse are installed
- An understanding of the basic concepts of relational database management
- Experience administering IBM DB2 Universal Database

Additionally, report designers and warehouse pack creators should have the following knowledge or experience:

- An understanding of the source data and application
- Data warehouse information and design, extract, transform, and load (ETL) processes, and online analytical processing (OLAP)

1.3 Publications

This section lists publications in the Tivoli Data Warehouse library and other related documents. It also describes how to access Tivoli publications online and how to order Tivoli publications.

The following sets of documentation are available to help you understand, install, and manage this warehouse pack:

- IBM Tivoli Monitoring for Transaction Performance
- Tivoli Data Warehouse

- IBM DB2 Universal Database, DB2 Data Warehouse Center, and DB2 Warehouse Manager
- IBM Redbooks

1.3.1 IBM Tivoli Monitoring for Transaction Performance library

The following documents are available on the Tivoli Software Information Center:

- *Tivoli Monitoring for Transaction Performance Installation Guide*, SC32-9107

Provides prerequisite information and instructions for installing the product. This guide also contains information that you might find useful after installing the product, such as uninstallation

- *IBM Tivoli Monitoring for Transaction Performance Evaluation Guide*, GC32-9190

Provides prerequisite information and instructions for performing a very simple installation of the product and deploying one of the components for evaluation purposes. This document enables you to go through the installation wizards and see the product's user interface performing some basic task and functions.

- *IBM Tivoli Monitoring for Transaction Performance Administrator's Guide*, GC32-9189

Provides detailed procedures for deploying and using each of the product components.

- *IBM Tivoli Monitoring for Transaction Performance Checking Performance and Availability Guide*, SC32-9106

Describes the browser-based graphical user interface (GUI), the help system, and how to produce graphical reports from transaction performance data.

- *IBM Tivoli Monitoring for Transaction Performance Problem Determination Guide*, SC32-9108

Provides the latest information about known product limitations and workarounds for the product. To ensure that the information is the latest available, this document is provided only on the Web, where it is updated as needed.

- *Program Directory for IBM Tivoli Monitoring for Transaction Performance for z/OS, version 5.3*

Describes installing and maintaining IBM Tivoli Monitoring for Transaction Performance for z\OS.

1.3.2 Tivoli Data Warehouse library

The following documents are available in the Tivoli Data Warehouse library. The library is available on the Tivoli Data Warehouse Documentation CD as well as online, as described in "Accessing publications online" on page 13.

- *Tivoli Data Warehouse Release Notes*, SC32-1399

Provides late-breaking information about Tivoli Data Warehouse and lists hardware requirements and software prerequisites.

- *Installing and Configuring Tivoli Data Warehouse*, GC32-0744

Describes how Tivoli Data Warehouse fits into your enterprise, explains how to plan for its deployment, and gives installation and configuration instructions. It contains maintenance procedures and troubleshooting information.

- *Enabling an Application for Tivoli Data Warehouse*, GC32-0745
Provides information about connecting an application to Tivoli Data Warehouse. This book is for application programmers who use Tivoli Data Warehouse to store and report on their application data, data warehousing experts who import Tivoli Data Warehouse data into business intelligence applications, and customers who put their local data in Tivoli Data Warehouse. This document is available only from the IBM Web site.
- *Tivoli Data Warehouse Messages*, SC09-7776
Lists the messages generated by Tivoli Data Warehouse, and describes the corrective actions you should take.

1.3.3 Related publications

The following sections describe additional publications to help you understand and use Tivoli Data Warehouse.

1.3.3.1 IBM Redbooks

IBM Redbooks are developed and published by the IBM International Technical Support Organization, the ITSO. They explore integration, implementation, and operation of realistic customer scenarios. The following Redbooks contain information about Tivoli Data Warehouse:

- *Introduction to Tivoli Enterprise Data Warehouse*, SG24-6607-00
Provides a broad understanding of Tivoli Data Warehouse. Some of the topics that are covered are concepts, architecture, writing your own extract, transform, and load processes (ETLs), and best practices in creating data marts.
- *Planning a Tivoli Enterprise Data Warehouse Project*, SG24-6608-00
Describes the necessary planning you must complete before you can deploy Tivoli Data Warehouse. The guide shows how to apply these planning steps in a real-life deployment of a warehouse pack using IBM Tivoli Monitoring. It also contains frequently used Tivoli and DB2 Universal Database commands and lists troubleshooting tips for Tivoli Data Warehouse.
- *End-to-End e-business Transaction Management Made Easy*, SG24-6080-00
- *Implementing Tivoli Data Warehouse 1.2*, SG24-7100
Focuses on planning, installation, customization, use, maintenance, and troubleshooting topics related to the new features of Tivoli Data Warehouse, Version 1.2. This is done using a number of case study scenarios and several warehouse packs.

1.3.3.2 IBM DB2 Universal Database, DB2 Data Warehouse Center, and DB2 Warehouse Manager library

The DB2 Universal Database library contains important information about the database and data warehousing technology provided by IBM DB2 Universal Database, DB2 Data Warehouse Center, and DB2 Warehouse Manager. Refer to the DB2 Universal Database library for help in installing, configuring, administering, and troubleshooting DB2 Universal Database, which is available on the IBM Web site:

<http://www-3.ibm.com/software/data/db2/library/>

After you install DB2 Universal Database, its library is also available on your system.

The following DB2 Universal Database documents are particularly relevant for people working with Tivoli Data Warehouse:

- *IBM DB2 Universal Database for Windows Quick Beginnings*, GC09-2971

Guides you through the planning, installation, migration (if necessary), and setup of a partitioned database system using the IBM DB2 product on Microsoft Windows.

- *IBM DB2 Universal Database for UNIX Quick Beginnings*, GC09-2970

Guides you through the planning, installation, migration (if necessary), and setup of a partitioned database system using the IBM DB2 product on UNIX.

- *IBM DB2 Universal Database Administration Guide: Implementation*, SC09-2944

Covers the details of implementing your database design. Topics include creating and altering a database, database security, database recovery, and administration using the Control Center, which is a DB2 graphical user interface.

- *IBM DB2 Universal Database Data Warehouse Center Administration Guide*, SC26-9993

Provides information on how to build and maintain a data warehouse using the DB2 Data Warehouse Center.

- *IBM DB2 Warehouse Manager Installation Guide*, GC26-9998

Provides information on how to install the following Warehouse Manager components: Information Catalog Manager, warehouse agents, and warehouse transformers.

- *IBM DB2 Universal Database and DB2 Connect Installation and Configuration Supplement*, GC09-2957

Provides advanced installation considerations, and guides you through the planning, installation, migration (if necessary), and set up of a platform-specific DB2 Universal Database client. This supplement also contains information on binding, setting up communications on the server, the DB2 GUI tools, DRDA® AS, distributed installation, the configuration of distributed requests, and accessing heterogeneous data sources.

- *IBM DB2 Universal Database Message Reference Volume 1*, GC09-2978 and *IBM DB2 Universal Database Message Reference Volume 2*, GC09-2979

Lists the messages and codes issued by DB2 Universal Database, the Information Catalog Manager, and the DB2 Data Warehouse Center, and describes the actions you should take.

1.3.4 Accessing publications online

The publications CD or product CD contains the publications that are in the product library. The format of the publications is PDF, HTML, or both.

IBM posts publications for this and all other Tivoli products, as they become available and whenever they are updated, to the Tivoli Software Information Center Web site. The Tivoli Software Information Center is located at the following Web address:

<http://publib.boulder.ibm.com/tividd/td/tdprodlist.html>

Note: If you print PDF documents on other than letter-sized paper, select the **Fit to page** check box in the Adobe Acrobat Print dialog. This option is available when you click **File → Print. Fit to page** ensures that the full dimensions of a letter-sized page print on the paper that you are using.

1.3.5 Ordering publications

You can order many Tivoli publications online at the following Web site:

<http://www.elink.ibm.com/public/applications/publications/cgibin/pbi.cgi>

You can also order by telephone by calling one of these numbers:

- In the United States: 800-879-2755
- In Canada: 800-426-4968

- In other countries, for a list of telephone numbers, see the following Web site:

<http://www.ibm.com/software/tivoli/order-lit/>

1.4 Accessibility

Accessibility features help users with a physical disability, such as restricted mobility or limited vision, to use software products successfully. For the warehouse pack, you use the interfaces of DB2 Universal Database and the reporting tool. See those documentation sets for accessibility information.

1.5 Contacting software support

If you have a problem with a Tivoli product, refer to the following IBM Software Support Web site:

<http://www.ibm.com/software/sysmgmt/products/support/>

If you want to contact customer support, see the IBM Software Support Guide at the following Web site:

<http://techsupport.services.ibm.com/guides/handbook.html>

The guide provides information about how to contact IBM Software Support, depending on the severity of your problem, and the following information:

- Registration and eligibility
- Telephone numbers, depending on the country in which you are located
- Information you must have before contacting IBM Software Support

1.6 Participating in newsgroups

User groups provide software professionals with a forum for communicating ideas, technical expertise, and experiences related to the product. They are located on the Internet, and are available using standard news reader programs. These groups are primarily intended for user-to-user communication, and are not a replacement for formal support. You can use News Readers like Netscape Messenger or Microsoft Outlook to view these newsgroups:

Tivoli Data Warehouse

<news://news.software.ibm.com/ibm.software.tivoli.enterprise-data-warehouse>

1.7 Typeface conventions

This guide uses the following typeface conventions:

Bold

- Lowercase commands and mixed case commands that are otherwise difficult to distinguish from surrounding text
- Interface controls (check boxes, push buttons, radio buttons, spin buttons, fields, folders, icons, list boxes, items inside list boxes, multicolumn lists, containers, menu choices, menu names, tabs, property sheets), labels (such as **Tip** and **Operating system considerations**)
- Column headings in a table
- Keywords and parameters in text

Italic

- Citations (titles of books, diskettes, and CDs)
- Words defined in text
- Emphasis of words (words as words)

- Letters as letters
- New terms in text (except in a definition list)
- Variables and values you must provide

Monospace

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

2 Overview

The following sections provide an overview of Tivoli Data Warehouse and the warehouse pack for IBM Tivoli Monitoring for Transaction Performance.

2.1 Overview of Tivoli Data Warehouse

Tivoli Data Warehouse provides the infrastructure for the following:

- Extract, transform, and load (ETL) processes through the IBM DB2 Data Warehouse Center tool
- Schema generation of the central data warehouse
- Historical reports

As shown in Figure 1, Tivoli Data Warehouse consists of a centralized data store where historical data from many management applications can be stored, aggregated, and correlated.

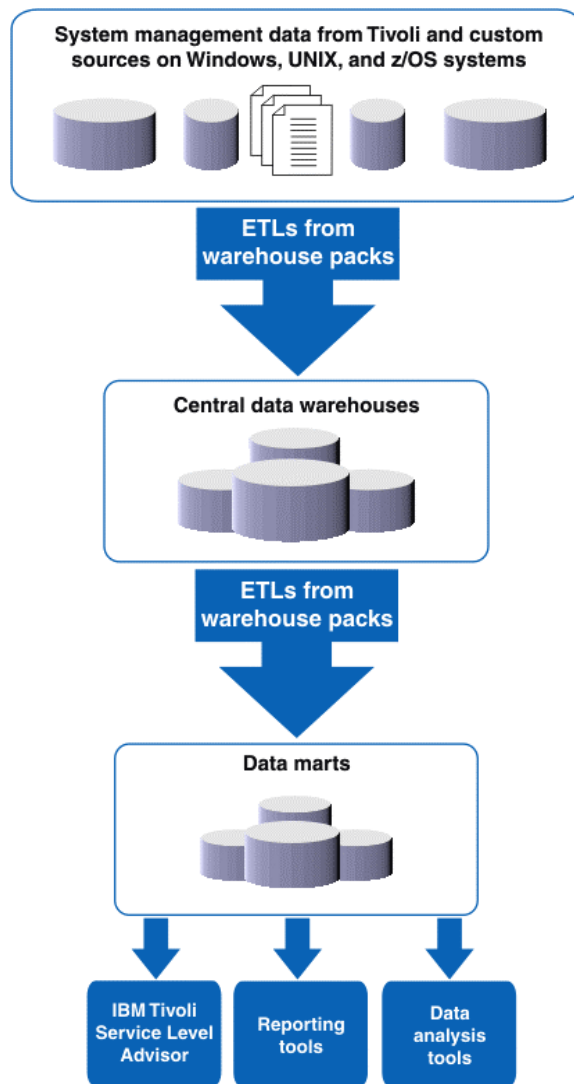


Figure 1. Tivoli Data Warehouse basic architecture

The *central data warehouse* uses a generic schema that is the same for all applications. As new components or new applications are added, more data is added to the database; however, no new database objects are added in the schema.

A *data mart* is a subset of a data warehouse that contains data that is tailored and optimized for the specific reporting needs of a department or team.

The *central data warehouse ETL* reads the data from the operational data stores of the application that collects it, verifies the data, makes the data conform to the schema, and places the data into the central data warehouse.

The *data mart ETL* extracts a subset of data from the central data warehouse, transforms it, and loads it into one or more star schemas, which can be included in data marts to answer specific business questions.

A program that provides these ETLs is called a *warehouse enablement pack* or simply *warehouse pack*.

The ETLs are typically scheduled to run periodically, usually during non-peak hours. If an ETL encounters data that it cannot correctly transform, it creates an entry in an exception table. Exception tables are described on page 118.

2.2 Overview of IBM Tivoli Monitoring for Transaction Performance Version 5.3 warehouse pack

IBM Tivoli Monitoring for Transaction Performance, Version 5.3 has the ability to display the transaction processing information for historical reporting. The data is stored in the customer's database that runs on either the DB2 Universal Database or Oracle database products. This database is regarded as the IBM Tivoli Monitoring for Transaction Performance, Version 5.3 *source database* for this warehouse pack.

Once the IBM Tivoli Monitoring for Transaction Performance, Version 5.3 real time reporting data is stored in the source database, the central data warehouse ETL process periodically (normally once a day) extracts data from the source database into the central data warehouse that is usually called TWH_CDW.(Version 1.2 supports more than one central data warehouse; if you have more than one central data warehouse, it may have a different name.) The central data warehouse ETL process converts the data into the IBM Tivoli Monitoring for Transaction Performance, Version 5.3 warehouse pack data model shown in Figure 2. This data model allows the IBM Tivoli Monitoring for Transaction Performance, Version 5.3 reporting data to fit into the general schema of Tivoli Data Warehouse, Version 1.2 for historical reporting and trending.

Note: Policy groups and agent groups will not have components in the CDW unless the measurement data extracted is associated with them to form a relationship.

The Tivoli Monitoring for Transaction Performance source database tables that provide data to the central data warehouse are as shown in Figure 2 below.

Table Name	Table data uploaded into the central data warehouse
EP	UUID1, HOSTNAME, IPADDRESS, DESCRIPTION
Transaction	TX_DESC, TX_NAME, TX_ID
Application	APPL_NAME, APPL_ID
Arm_User	USER_ID, USER_NAME
Node	NODE_ID, APPL_ID, TX_ID, USER_ID, HOST_ID, NODE_SEQUENCE, EP_OM_ID
Aggregatedata	THRESH_VIOL_CNT, BAD_STATUS_CNT, ABORT_STATUS_CNT, COUNT, AVERAGE_VALUE, MAXIMUM_VALUE, MINIMUM_VALUE, AGGREGATE_ID, RELATIONMAP_ID, COLL_DATETIME, PARTIAL, PATT_TRANS_OM_ID, COLL_TIME_SPAN

Relationmap	RELATIONMAP_ID, MGMT_POLICY_OM_ID, PARENT_NODE_ID, CURRENT_NODE_ID, ROOT_NODE_ID, RELATED_NODE_ID
PT	UUID1
Threshold	UUID1, THRPTINVERSE_UUID, THRESHOLDVALUE, TR_UUID
TR	UUID1, MANAGEMENTPOLICYID, RESPONSELEVEL
EPG	UUID1, GROUPNAME, DESCRIPTION
EpEgpInverse_Ep	EpEGPInverse_UUID, EP_UUID
Managementpolicy	NAME, DESCRIPTION, OBJECTVERSION, CREATOR, CREATED, DELETED, DELETOR, LASTUPDATED, UPDATOR, DESCRIPTION, UUID1, ISDELETE, TYPE1, STATE1, EDGEPOLICY_UUID, SCHEDULE_UUID, ENDPOINTGROUP_UUID
MPolicies_PGps	PGPs_UUID, MPolicies_UUID
PolicyGroup	UUID1, NAME, DESCRIPTION

Figure 2. IBM Tivoli Monitoring for Transaction Performance database source tables

After the central data warehouse ETL process completes, the data mart ETL process loads data from the central data warehouse into the data mart. The data mart, fact tables, dimension tables, and helper tables are created in the BWM schema. The data from the central data warehouse is used to populate the dimension and fact tables in the BWM data mart. You can then utilize the hourly, daily, weekly, and monthly star schemas of the dimension and fact tables to generate reports using Crystal Reports.

IBM Tivoli Monitoring for Transaction Performance does not use resource models, thus the IBM Tivoli Monitoring warehouse pack and its tables are not required by this warehouse pack.

Figure 3 shows the supported components and their relationships for IBM Tivoli Monitoring for Transaction Performance.

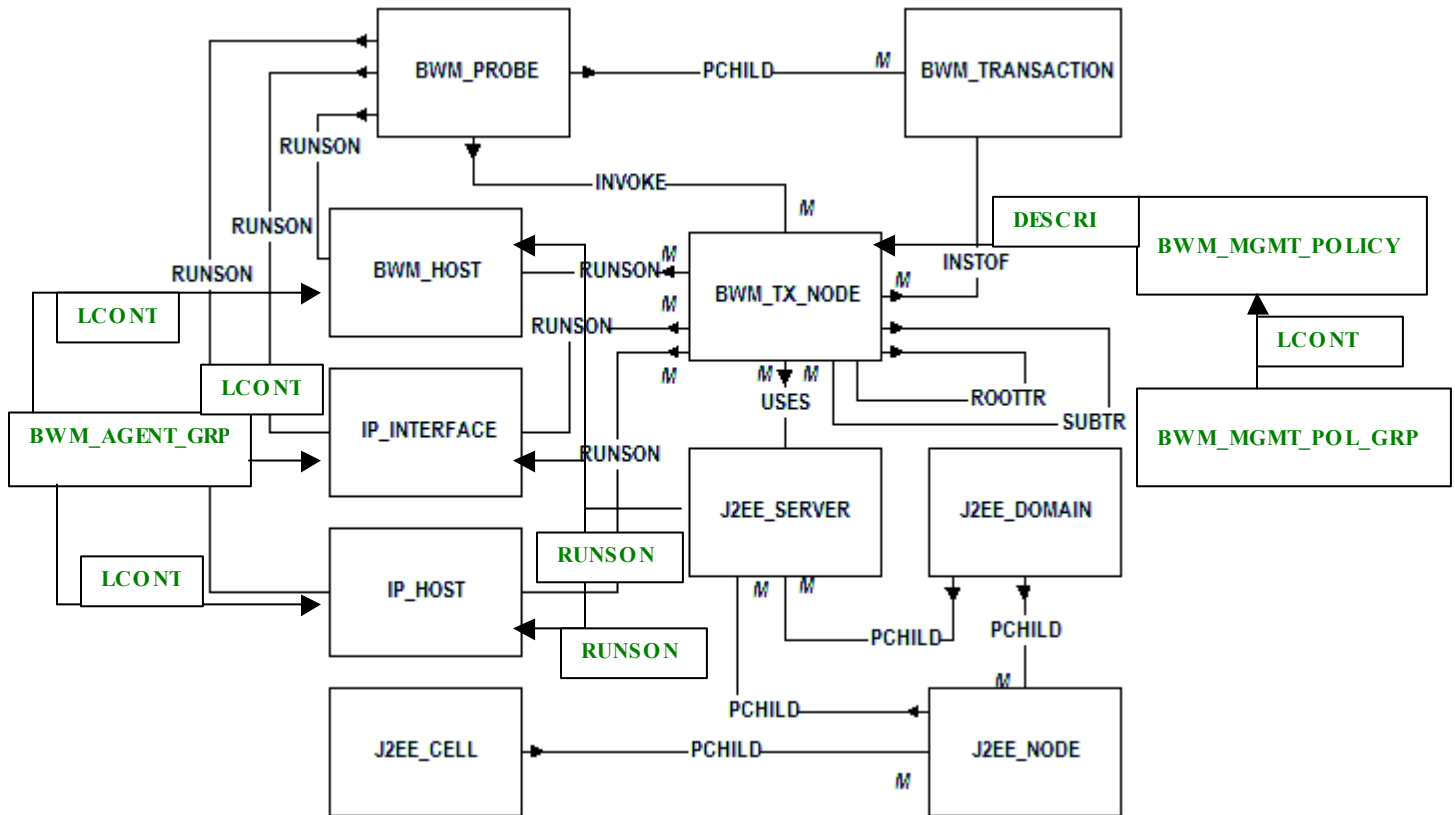


Figure 3, IBM Tivoli Monitoring for Transaction Performance, Version 5.3 warehouse pack component data model

The following table contains a short description of the components used in IBM Tivoli Monitoring for Transaction Performance. See the Component relationship rule table in section 7.1.4 for details on component relationships for IBM Tivoli Monitoring for Transaction Performance.

Component Name	Component Type Code	Description of Component
Transaction Node	BWM_TX_NODE	A transaction node is representative of a unique host, application, transaction, and user combination and should be considered a unique representation of a transaction or sub-transaction. If the transaction name has a length of more than 254 characters then the corresponding transaction name will be stored in the BWM.COMP_NAME_LONG table and the name of the node in the Comp table will be the first 240 characters of the transaction name.
Transaction	BWM_TRANSACTION	A transaction represents a business process that is identified by the transaction name.
Monitoring Probe	BWM_PROBE	The monitoring probe represents an application that creates synthetic transaction for monitoring purposes (STI), an application that monitors transactions for quality of service purposes (QOS), an application that records user interaction to collect timing metrics (Generic Windows), and ARM instrumented applications (ARM).

Transaction Host	BWM_HOST	The transaction host represents the machine or IP host on which a transaction runs. The transaction host can have values of an IP address (IP_INTERFACE), a fully qualified host name (IP_HOST), or a short host name (BWM_HOST).
Agent Group	BWM_AGENT_GRP	The agent group is a group of hosts defined as belonging to the same logical agent group for monitoring purposes.
Management Policy	BWM_MGMT_POLICY	The name of the management policy that a transaction is defined to.
Management Policy Group	BWM_MGMT_POL_GRP	The management policy group is a grouping of management policies defined as belonging to the same logical management policy group for monitoring purposes.
J2EE Server	J2EE_SERVER	The J2EE server component represents a J2EE Web Application Server. In this release of IBM Tivoli Monitoring for Transaction Performance, we support WebSphere 5.0 (Cell/Node/Application Server), WebSphere 4.1 (Node/Application Server), and WebLogic 7.0.1 (Domain/Application Server).
J2EE Node	J2EE_NODE	A J2EE node represents the machine upon which J2EE components run. Weblogic does not have a node, only Websphere does.
J2EE Domain	J2EE_DOMAIN	A J2EE domain represents the specified range of J2EE managed objects. This component applies to WebSphere 4.x.
J2EE Cell	J2EE_CELL	A J2EE cell is a grouping of J2EE Nodes into a single administrative domain. This component applies to WebSphere 5.0 and up.

3 Reports

This section provides information about the predefined reports provided by the warehouse pack.

The following information is provided:

- A list of the reports
- A description of the information contained in the report
- Sample report layouts
- The name of the table that is used to create the report
- A section that describes how the report input parameters work and the criteria for retrieving data for the report

The following table in Figure 4 shows the information for each predefined report. An example of each report is shown in section 3.1, Sample report layouts.

Report name	Description	Table names
Average response time by application	This line graph and detail report show individual application average response times across all hosts for the specified time period. Application response time is the average of the transaction response times for all transactions defined within that application. The transaction response time measurement unit is in seconds. Input parameters define the time period's start and end times, applications to show and from which fact table to get the data. The granularity of the report will be dependant on the fact table chosen to retrieve data from. This report aids in determining which applications have longer response times and helps pinpoint which applications need improvement. This report utilizes the BWM Daily Transaction Node Star Schema.	Fact tables used: BWM.F_TX_ND_HOUR BWM.F_TX_ND_DAY BWM.F_TX_ND_WEEK BWM.F_TX_ND_MONTH Other data mart tables used are: BWM.D_TX_ND_METRIC BWM.D_APP
Average response time by user for Transaction <transaction>	This line graph and detail report show response times across all hosts, for each user for a particular edge transaction over the time period set by an input parameter. User transaction response time is the average of the edge transactions' response times for the user and time period specified. The transaction and users shown, as well as from which fact table to get the data, are determined by input parameters. The granularity of the	Fact tables used: BWM.F_TX_ND_HOUR BWM.F_TX_ND_DAY BWM.F_TX_ND_WEEK BWM.F_TX_ND_MONTH Other data mart tables used are:

	parameters. The granularity of the report will be dependant on the fact table chosen to retrieve data from. The transaction response time measurement unit is seconds. This report indicates the overall performance of the user's specified edge transaction. This can aid in determining the overall transaction response time for many users to see if all users running the same transaction experience the same kind of response time. This report utilizes the BWM Daily Transaction Node Star Schema.	BWM.D_TX_ND_METRIC BWM.D_TX BWM.D_TX_ND
Average response time by policy	This line graph and detail report show response times across all hosts, for each monitoring policy, over the time period set by an input parameter. Policy transaction response time is the average of all transactions' response times for that policy for that time period. The policies shown are set by an input parameter and from which fact table to get the data. The granularity of the report will be dependant on the fact table chosen to retrieve data from. The transaction response time measurement unit is seconds. This report indicates the relative performance of transactions for particular policies. This can help compare the response times of the policies shown to determine if the host groupings in the policy should be modified. This report utilizes the BWM Daily Transaction Node Star Schema.	Fact tables used: BWM.F_TX_ND_HOUR BWM.F_TX_ND_DAY BWM.F_TX_ND_WEEK BWM.F_TX_ND_MONTH Other data mart tables used are: BWM.D_TX_ND_METRIC BWM.D_TX_ND
Average response time by policy group	This stacked bar graph and detail report show average response times for policy groups averaged over the desired time period. Input parameters determine the time period and policy groups shown and from which fact table to get the data. The granularity of the report will be dependant on the fact table chosen to retrieve data from. Policy group transaction response time is the average of all transactions' response times for that policy group for the time period specified. The transaction response time measurement unit is seconds. This report demonstrates, at a high level, the performance of all transactions for	Fact tables used: BWM.F_TX_ND_HOUR BWM.F_TX_ND_DAY BWM.F_TX_ND_WEEK BWM.F_TX_ND_MONTH Other data mart tables used are: BWM.D_TX_ND_METRIC BWM.D_TX_ND_POL_GRP

	<p>a policy group. It can help to determine whether transactions' overall response time differs by policy groups so any significant differences can be investigated and remedied. This report utilizes the BWM Daily Transaction Node Star Schema.</p>	
<p>Average response time by policy for policy group <policy_group></p>	<p>This line graph and detail report show average response times for all the policies of a policy group, for the desired time period. Input parameters determine the time period and policy group shown and from which fact table to get the data. The granularity of the report will be dependant on the fact table chosen to retrieve data from. Policy transaction response time is the average of all transactions' response times for that policy for the time period specified. The transaction response time measurement unit is seconds. This report demonstrates, at a high level, the performance of all transactions for all the policies in a policy group. It can help to determine whether transactions' overall response time differs among the policies defined to a policy groups so any significant differences can be investigated and corrected. This report utilizes the BWM Daily Transaction Node Star Schema.</p>	<p>Fact tables used:</p> <p>BWM.F_TX_ND_HOUR BWM.F_TX_ND_DAY BWM.F_TX_ND_WEEK BWM.F_TX_ND_MONTH</p> <p>Other data mart tables used are:</p> <p>BWM.D_TX_ND_METRIC BWM.D_TX_ND BWM.D_TX_ND_POL_GRP</p>
<p>Average response time by agent group for transaction <transaction></p>	<p>This line graph and detail report show transaction response times for the agent groups specified, for the desired time period. Input parameters determine the time period and agent groups shown and from which fact table to get the data. The granularity of the report will be dependant on the fact table chosen to retrieve data from. Agent group transaction response time is the average of all transactions' response times for that agent group's hosts for the time period specified. The transaction response time measurement unit is seconds. This report demonstrates, at a high level, the performance of all transactions on an agent group. It can help to determine whether transactions' overall response time has been due to any particular agent group, which would indicate a resource issue in that</p>	<p>Fact tables used:</p> <p>BWM.F_TX_ND_HOUR BWM.F_TX_ND_DAY BWM.F_TX_ND_WEEK BWM.F_TX_ND_MONTH</p> <p>Other data mart tables used are:</p> <p>BWM.D_TX_ND_METRIC BWM.D_TX BWM.D_HOST_AGENT_GRP</p>

	group. This report utilizes the BWM Daily Transaction Node Star Schema.	
Execution load by agent group for Transaction <transaction>	<p>This line graph and detail report show a transaction's execution count for each agent group averaged over the desired time period. These counts do not represent actual transaction counts unless the policy sampling rates of the transaction has been set to 100%. Otherwise, it shows proportionally how many times each user is running the specified edge transaction. Input parameters determine the time period and the agent groups shown and from which fact table to get the data. The granularity of the report will be dependant on the fact table chosen to retrieve data from. The transaction counts include successful and failed transaction executions. This report indicates the load that each agent group was under for a given timeframe. It can show unbalanced loads or agent group servers that need additional resources. This report utilizes the BWM Daily Transaction Node Star Schema.</p>	<p>Fact tables used:</p> <p>BWM.F_TX_ND_HOUR BWM.F_TX_ND_DAY BWM.F_TX_ND_WEEK BWM.F_TX_ND_MONTH</p> <p>Other data mart tables used are:</p> <p>BWM.D_TX_ND_METRIC BWM.D_TX BWM.D_HOST_AGENT_GRP</p>
Execution load by application	<p>This line graph and detail report show the number of transactions that executed for a particular application across all hosts for the time period specified. These counts do not represent actual transaction counts unless the policy sampling rates of the transaction has been set to 100%. Otherwise, it shows proportionally how many times each user is running the specified edge transaction. Input parameters determine the time period and the applications shown and from which fact table to get the data. The granularity of the report will be dependant on the fact table chosen to retrieve data from. The transaction counts include successful and failed transaction executions. This report indicates the transaction load that each application type generated for the given timeframe. If an application has an unusually low number of transactions running for it, the application may not have been available or used during the interval. This report utilizes the BWM Daily</p>	<p>Fact tables used:</p> <p>BWM.F_TX_ND_HOUR BWM.F_TX_ND_DAY BWM.F_TX_ND_WEEK BWM.F_TX_ND_MONTH</p> <p>Other data mart tables used are:</p> <p>BWM.D_TX_ND_METRIC BWM.D_APP</p>

	Transaction Node Star Schema.	
Execution load by user for transaction <transaction>	<p>This line graph and detailed report show the number of times the specified edge transaction was run by particular users over a given timeframe. The detail data shows the actual counts. These counts do not represent actual transaction counts unless the policy sampling rates of the transaction has been set to 100%. Otherwise, it shows proportionally how many times each user is running the specified edge transaction. Input parameters determine the transaction, time period and users shown and from which fact table to get the data. The granularity of the report will be dependant on the fact table chosen to retrieve data from. The transaction counts include successful and failed transaction executions. This report indicates the transaction load that each user places on the systems for a particular edge transaction. This report utilizes the BWM Daily Transaction Node Star Schema.</p>	<p>Fact tables used:</p> <p>BWM.F_TX_ND_HOUR BWM.F_TX_ND_DAY BWM.F_TX_ND_WEEK BWM.F_TX_ND_MONTH</p> <p>Other data mart tables used are:</p> <p>BWM.D_TX_ND_METRIC BWM.D_TX_ND BWM.D_TX</p>
Execution load for policy <policy>	<p>This line graph and detail report shows the number of times that transactions, monitored by policies, have executed across all hosts during the time period specified. The time period and policy to show are specified by input parameters and from which fact table to get the data. The granularity of the report will be dependant on the fact table chosen to retrieve data from. The transaction counts include successful and failed transaction executions. This report gives an indication of the transaction load for the specified policies monitored over time. The indicated load is not the true load on the server unless the policy creator set the sampling rate to 100%. For partial sampling rates, this graph shows the relative loads, over time, for a policy and can be used to determine peak times of usage for a policy's transactions. This report utilizes the BWM Daily Transaction Node Star Schema.</p>	<p>Fact tables used:</p> <p>BWM.F_TX_ND_HOUR BWM.F_TX_ND_DAY BWM.F_TX_ND_WEEK BWM.F_TX_ND_MONTH</p> <p>Other data mart tables used are:</p> <p>BWM.D_TX_ND_METRIC BWM.D_TX_ND</p>

Execution load by Agent for policy <policy>	<p>This line graph and detail report show the number of times that transactions, monitored by a given policy, have executed across all hosts defined to the specified policy during the time period chosen. The time period and policy to show are specified by input parameters and from which fact table to get the data. The granularity of the report will be dependant on the fact table chosen to retrieve data from. The transaction counts include successful and failed transaction executions. This report gives an indication of the transaction load that a given policy monitored over time. The indicated load is not the true load on the server unless the policy creator set the sampling rate to 100%. For partial sampling rates, this graph shows the relative loads, over time, for a policy and can be used to determine peak times of usage for a policy's transactions. This report utilizes the BWM Daily Transaction Node Star Schema.</p>	<p>Fact tables used:</p> <p>BWM.F_TX_ND_HOUR</p> <p>BWM.F_TX_ND_DAY</p> <p>BWM.F_TX_ND_WEEK</p> <p>BWM.F_TX_ND_MONTH</p> <p>Other data mart tables used are:</p> <p>BWM.D_TX_ND_METRIC</p> <p>BWM.D_TX_ND</p> <p>BWM.D_HOST</p>
Transaction availability for transaction <transaction>	<p>This line graph shows the percentage availability of a transaction across its executing endpoints. The transaction data to show is specified by an input parameter and from which fact table to get the data. The granularity of the report will be dependant on the fact table chosen to retrieve data from. Dips in the line graph below 100% indicate transaction failures that need to be investigated. This report gives an indication of when and on which agents a transaction was not 100% available. This report utilizes the BWM Daily Transaction Node Star Schema.</p>	<p>Fact tables used:</p> <p>BWM.F_TX_ND_HOUR</p> <p>BWM.F_TX_ND_DAY</p> <p>BWM.F_TX_ND_WEEK</p> <p>BWM.F_TX_ND_MONTH</p> <p>Other data mart tables used are:</p> <p>BWM.D_TX_ND_METRIC</p> <p>BWM.D_HOST</p> <p>BWM.D_TX</p>
Overall transaction response time for transaction <transaction>	<p>This line graph shows the average response time over time of the specified transaction for each agent it ran on. Response time and agent are specified by input parameters and from which fact table to get the data. The granularity of the report will be dependant on the fact table chosen to retrieve data from. This report indicates the relative performance of a transaction over each agent for</p>	<p>Fact tables used:</p> <p>BWM.F_TX_ND_HOUR</p> <p>BWM.F_TX_ND_DAY</p> <p>BWM.F_TX_ND_WEEK</p> <p>BWM.F_TX_ND_MONTH</p> <p>Other data mart tables used</p>

	comparison purposes. This report utilizes the BWM Daily Transaction Node Star Schema.	are: BWM.D_TX_ND_METRIC BWM.D_TX BWM.D_HOST
Policy Load per agent	This report shows the total number of policies that agents had running on them for a given timeframe. The time period and agents to show are specified by input parameters and from which fact table to get the data. The granularity of the report will be dependant on the fact table chosen to retrieve data from. This report shows the monitoring load, in terms of configured policies, across an infrastructure. It can also indicate the agents that are most monitored. This report utilizes the BWM Daily Transaction Node Star Schema.	Fact tables used: BWM.F_TX_ND_HOUR BWM.F_TX_ND_DAY BWM.F_TX_ND_WEEK BWM.F_TX_ND_MONTH Other data mart tables used are: BWM.D_TX_ND_METRIC BWM.D_HOST BWM.D_TX_ND
Least throughput	This report shows the top N transactions with the lowest throughput. Throughput is defined as the number of transactions executed for the specified time period for differing values of transaction, user, agent and application. The fact table from which to get the data, time period, transactions, number of transactions, users, applications and agents to show in the report is specified by input parameters. The granularity of the report will be dependant on the fact table chosen to retrieve data from. This report shows the least number of transactions executed by the specified set of transactions, users, applications and hosts. This report utilizes the BWM Daily Transaction Node Star Schema.	Fact tables used: BWM.F_TX_ND_HOUR BWM.F_TX_ND_DAY BWM.F_TX_ND_WEEK BWM.F_TX_ND_MONTH Other data mart tables used are: BWM.D_TX_ND_METRIC BWM.D_APP BWM.D_HOST BWM.D_TX_ND BWM.D_TX
Slowest transactions	This extreme case report shows the top N edge transactions that were the slowest transactions over the time period specified for the transactions, users, applications and hosts chosen. The time period, transactions, number of transactions for top N, users, applications, agents and from which fact table to get the data are specified	Fact tables used: BWM.F_TX_ND_HOUR BWM.F_TX_ND_DAY BWM.F_TX_ND_WEEK BWM.F_TX_ND_MONTH

	by input parameters. The granularity of the report will be dependant on the fact table chosen to retrieve data from. This report shows the worst performing transactions monitored for a set of transactions, users, applications and hosts. This report utilizes the BWM Daily Transaction Node Star Schema.	Other data mart tables used are: BWM.D_TX_ND_METRIC BWM.D_APP BWM.D_HOST BWM.D_TX_ND BWM.D_TX
Enhanced Application Availability	This stacked bar chart and detail report compare, at a high level, the number of threshold violations, successful, failed and aborted transactions across all hosts. The stacked bars allow for quick visual perception of the number of successful transactions to unsuccessful and aborted transactions as well as the threshold violations total count. The time period to show is specified by an input parameter and from which fact table to get the data. The granularity of the report will be dependant on the fact table chosen to retrieve data from. This report shows the success and failure rate for transaction execution and where the number of failures, threshold violations or aborted transactions needs further investigation. This report utilizes the BWM Daily Transaction Node Star Schema.	Fact tables used: BWM.F_TX_ND_HOUR BWM.F_TX_ND_DAY BWM.F_TX_ND_WEEK BWM.F_TX_ND_MONTH Other data mart tables used are: BWM.D_TX_ND_METRIC BWM.D_APP
Resource usage response time and execution load for transaction <transaction>	There are two graphs for this report. The first is a line graph showing the average transaction response time for the specified transaction for each monitored agent in the transaction path for the specified time period. The second is a combination bar and line graph showing the transaction execution load for the specified transaction for each monitored agent in the transaction path for the specified time period. For comparison, the transaction response time is charted against the average percent CPU and memory utilization on the same agent. This report is a cross-application report using transaction response time data from the Tivoli Monitoring for Transaction	Fact tables used: BWM.F_TX_ND_HOUR BWM.F_TX_ND_DAY BWM.F_TX_ND_WEEK BWM.F_TX_ND_MONTH AMY.F_CPU_HOUR AMY.F_CPU_DAY AMY.F_CPU_WEEK AMY.F_CPU_MONTH AMY.F_MEM_HOUR AMY.F_MEMORY_DAY AMY.F_MEMORY_WEEK

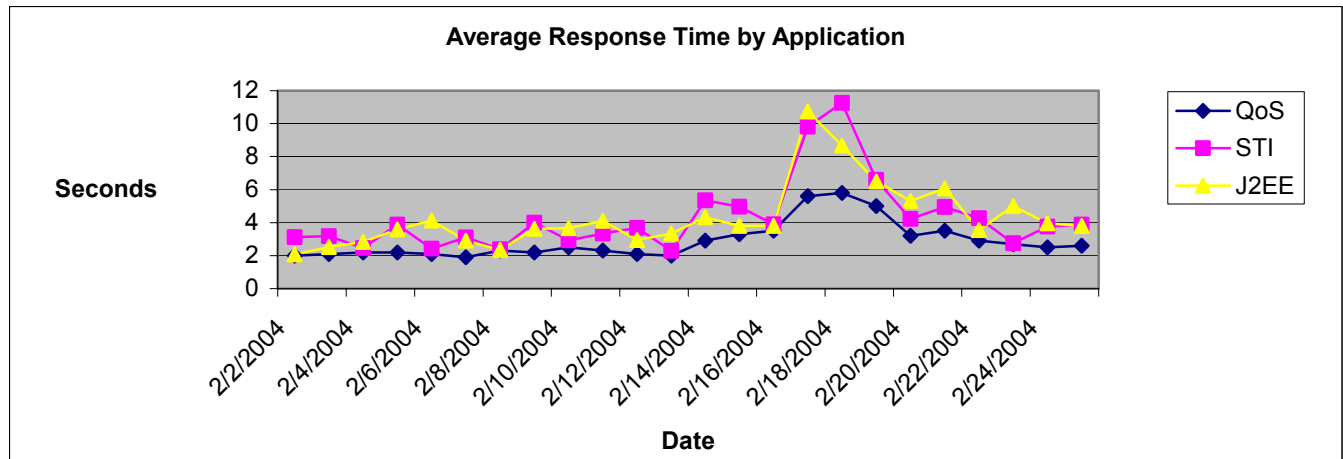
	<p>Performance warehouse pack and host memory and CPU utilization from the IBM Tivoli Monitoring warehouse pack. Average transaction response time is the average of all transactions' response times in the edge transaction's path for the agents the transactions ran on, for the time period specified. The transaction and time period shown are determined by input parameters, and from which fact table to get the data. The granularity of the report will be dependant on the fact table chosen to retrieve data from. The transaction response time measurement unit is seconds. This report demonstrates the load and performance of the transaction on the monitored agents in the transaction path. It can correlate transaction response time and execution load to host resource utilization to aid in determining whether the transaction's overall response time has been due to any resource issue on the agents. This report utilizes the BWM Daily Transaction Node Star Schema.</p> <p>Note: This report can only be run if the IBM Tivoli Monitoring AMX/AMY warehouse packs are installed in the same data mart database as this warehouse pack and the data for both packs contains the same time periods and many of the same hosts. In addition the script, <code>Create_Cross_App_Report_Views.sql</code>, in the <code>misc/tools</code> directory must have been run to create the required data views.</p>	<p>AMY.F_MEMORY_MONTH</p> <p>Other data mart tables used are:</p> <p>BWM.D_TX_ND_METRIC</p> <p>BWM.D_TX</p> <p>BWM.D_HOST</p> <p>AMY.D_HOST_IP</p> <p>AMY.D_CPU_METRIC</p> <p>AMY.D_MEM_METRIC</p>
--	--	--

Figure 4. IBM Tivoli Monitoring for Transaction Performance, Version 5.3 warehouse pack Predefined reports

3.1 Sample report layouts

The following are samples of the reports provided.

3.1.1 Average Response Time by Application



Average Response Time by Application

Application Name**QoS**

Date	Average Response Time (Seconds)
2/2/2004	2.1
2/3/2004	1.9
2/4/2004	2.0
2/5/2004	2.2
2/6/2004	2.1
Average over period	2.1

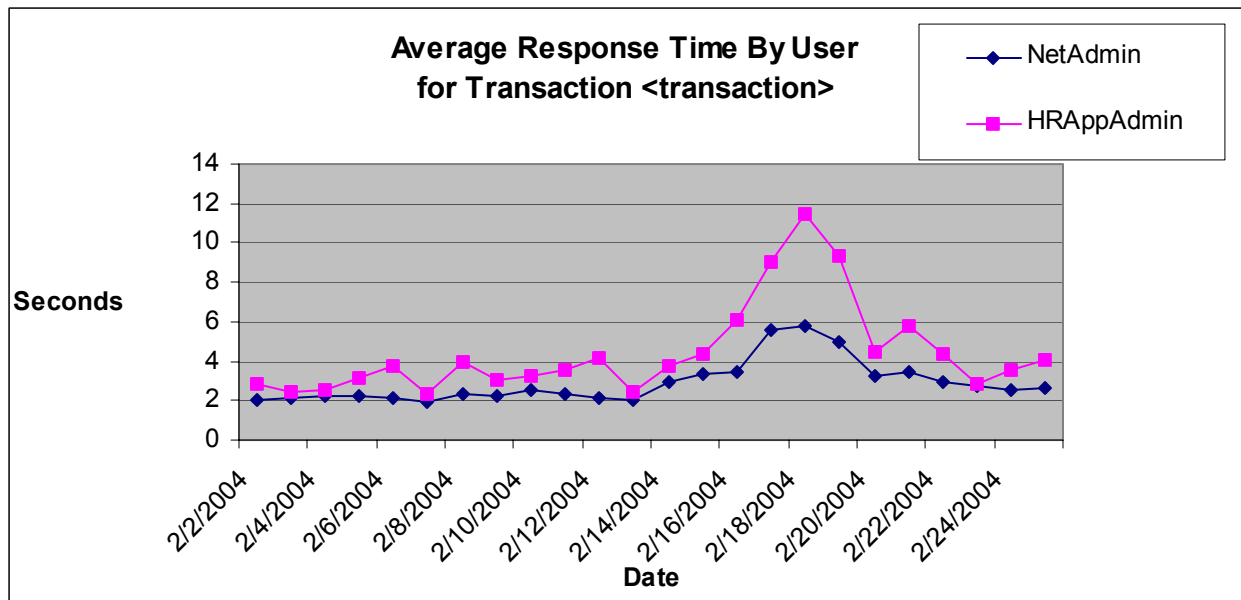
Application Name**STI**

Date	Average Response Time (Seconds)
2/2/2004	4.0
2/3/2004	2.2
2/4/2004	4.1
2/5/2004	2.1
2/6/2004	3.0
Average over period	3.1

Application Name**J2EE**

Date	Average Response Time (Seconds)
2/2/2004	3.0
2/3/2004	3.2
2/4/2004	2.1
2/5/2004	1.1
2/6/2004	4.0
Average over period	2.7

3.1.2 Average Response Time by User for Transaction <transaction>



**Average Response Time by User
For Transaction <transaction>**

User Name

NetAdmin

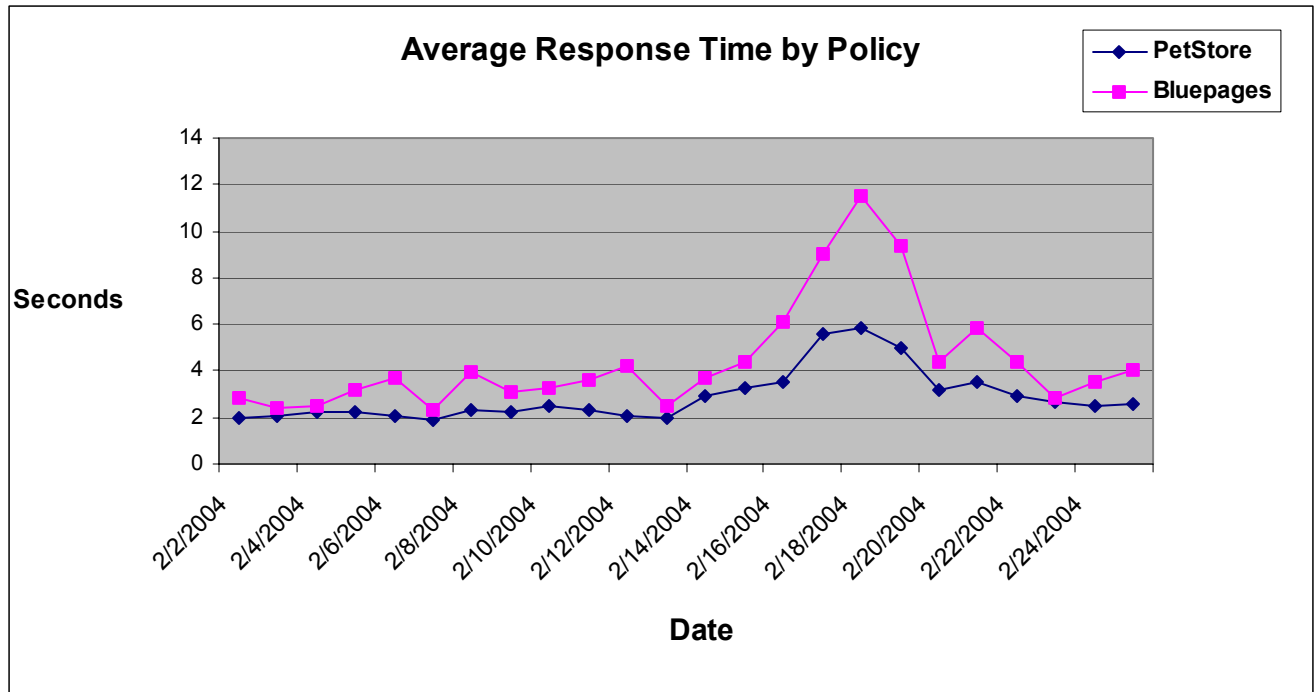
Date	Average Response Time (Seconds)
2/2/2004	2.1
2/3/2004	1.9
2/4/2004	2.0
2/5/2004	2.2
2/6/2004	2.1
Average over period	2.1

User Name

HRAppAdmin

Date	Average Response Time (Seconds)
2/2/2004	4.0
2/3/2004	2.2
2/4/2004	4.1
2/5/2004	2.1
2/6/2004	3.0
Average over period	3.1

3.1.3 Average Response Time by Policy

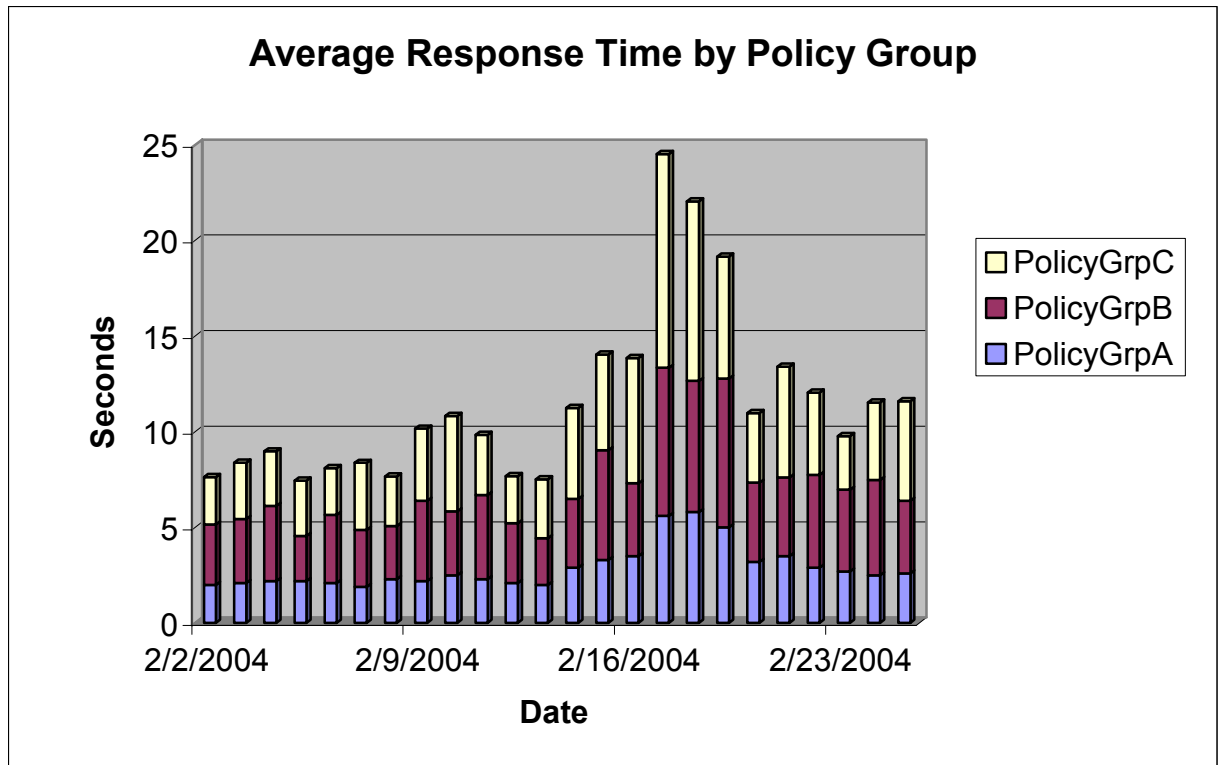
**Average Response Time by Policy****Policy Name****PetStore**

Date	Average Response Time (Seconds)
2/2/2004	2.1
2/3/2004	1.9
2/4/2004	2.0
2/5/2004	2.2
2/6/2004	2.1
Average over period	2.1

Policy Name**Bluepages**

Date	Average Response Time (Seconds)
2/2/2004	4.0
2/3/2004	2.2
2/4/2004	4.1
2/5/2004	2.1
2/6/2004	3.0
Average over period	3.1

3.1.4 Average Response Time by Policy Group



Average Response Time by Policy Group

Policy Group Name**PolicyGrpA**

Date	Average Response Time (Seconds)
2/2/2004	2.1
2/3/2004	1.9
2/4/2004	2.0
2/5/2004	2.2
2/6/2004	2.1
Average over period	2.1

Policy Group Name**PolicyGrpB**

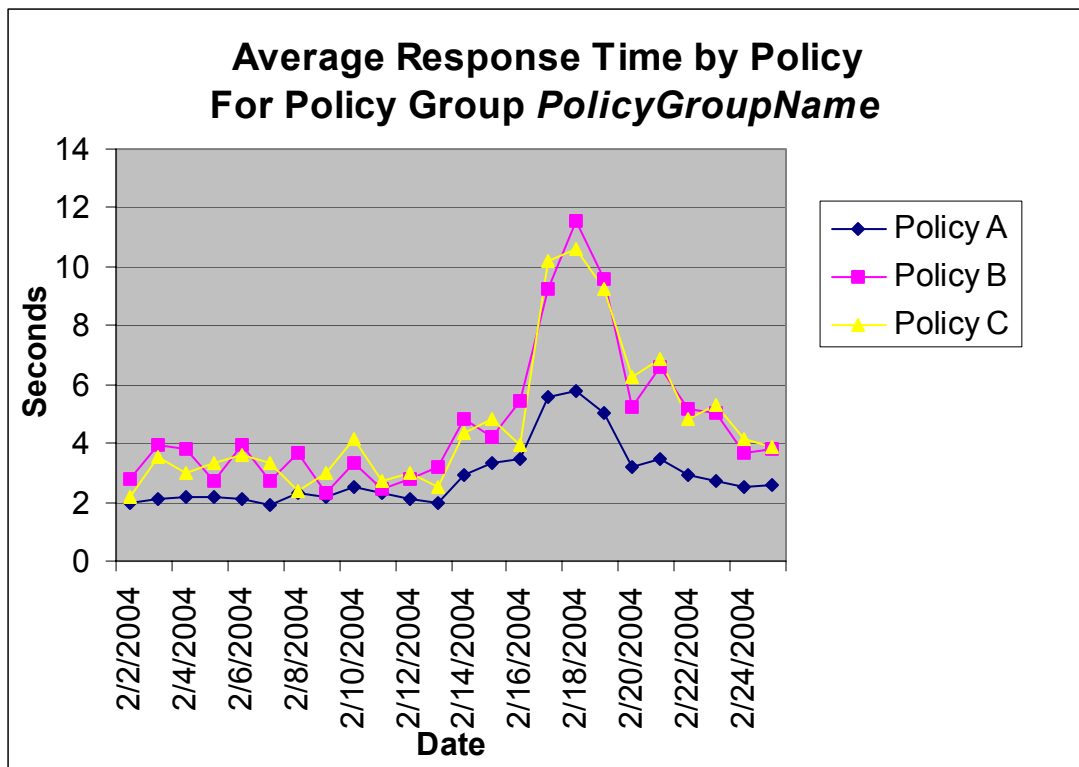
Date	Average Response Time (Seconds)
2/2/2004	4.0
2/3/2004	2.2
2/4/2004	4.1
2/5/2004	2.1
2/6/2004	3.0
Average over period	3.1

Policy Group Name**PolicyGrpC**

Date	Average Response Time (Seconds)
2/2/2004	3.0
2/3/2004	3.2

2/4/2004	2.1
2/5/2004	1.1
2/6/2004	4.0
Average over period	2.7

3.1.5 Average Response Time by Policy for Policy Group *PolicyGroupName*



Average Response Time by Policy For Policy Group *PolicyGroupName*

Policy Name

Policy A

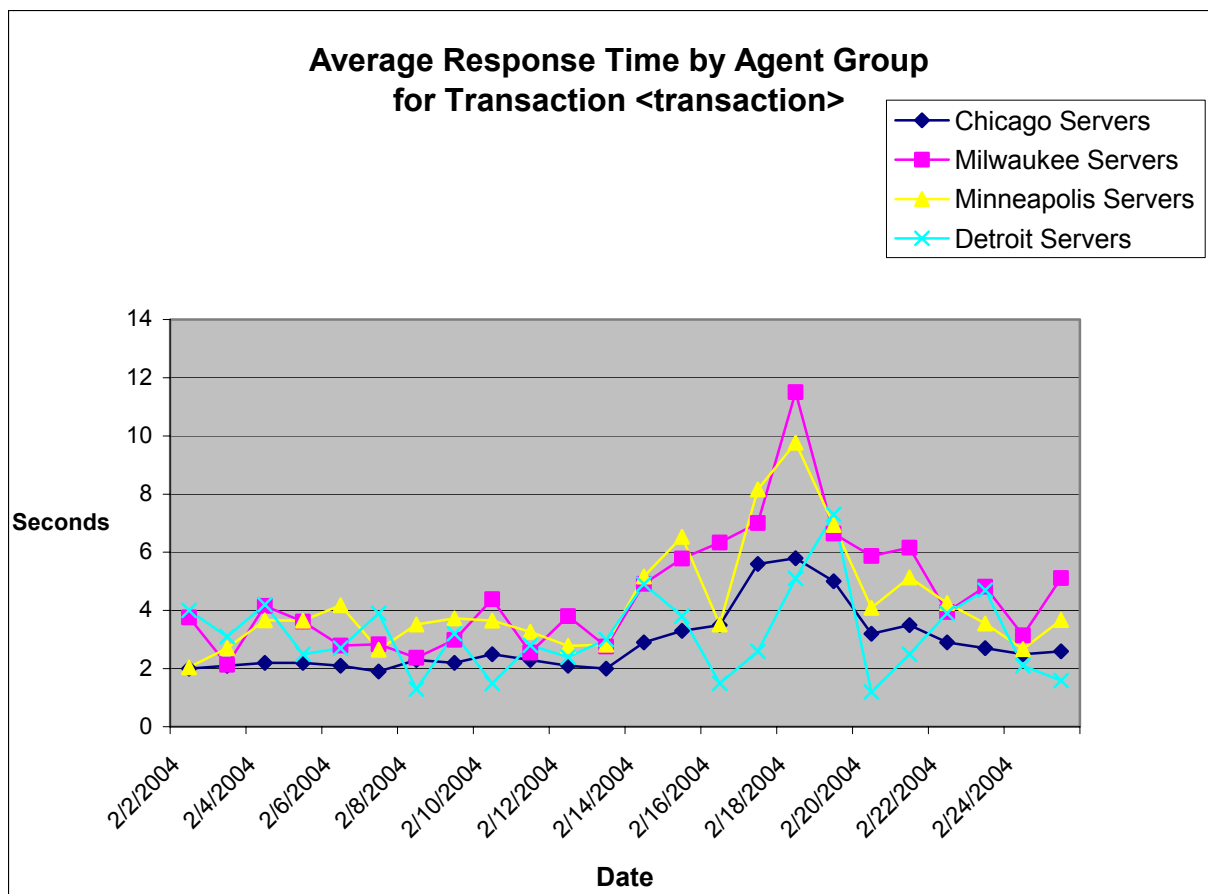
Date	Average Response Time (Seconds)
2/2/2004	2.1
2/3/2004	1.9
2/4/2004	2.0
2/5/2004	2.2
2/6/2004	2.1
Average over period	2.1

Policy Name

Policy B

Date	Average Response Time (Seconds)
2/2/2004	4.0
2/3/2004	2.2
2/4/2004	4.1
2/5/2004	2.1
2/6/2004	3.0
Average over period	3.1
Policy Name	
Policy C	
Date	Average Response Time (Seconds)
2/2/2004	3.0
2/3/2004	3.2
2/4/2004	2.1
2/5/2004	1.1
2/6/2004	4.0
Average over period	2.7

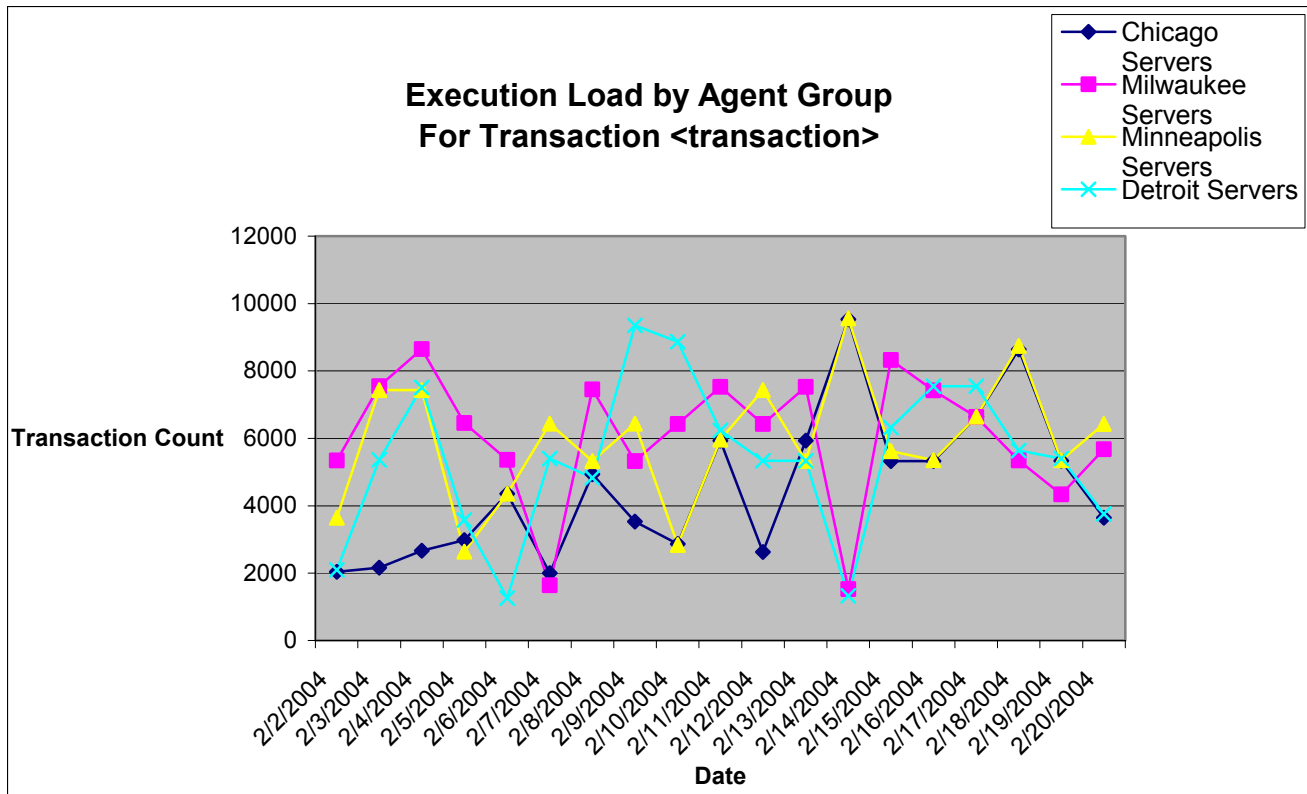
3.1.6 Average Response Time by Agent Group for Transaction <transaction>



**Average Response Time by Agent Group
For Transaction <transaction>**

Agent Group Name	
Chicago Servers	
Date	Average Response Time (Seconds)
2/2/2004	2.1
2/3/2004	1.9
2/4/2004	2.0
2/5/2004	2.2
2/6/2004	2.1
Average over period	2.1
Agent Group Name	
Milwaukee Servers	
Date	Average Response Time (Seconds)
2/2/2004	4.0
2/3/2004	2.2
2/4/2004	4.1
2/5/2004	2.1
2/6/2004	3.0
Average over period	3.1
Agent Group Name	
Minneapolis Servers	
Date	Average Response Time (Seconds)
2/2/2004	3.0
2/3/2004	3.2
2/4/2004	2.1
2/5/2004	1.1
2/6/2004	4.0
Average over period	2.7
Agent Group Name	
Detroit Servers	
Date	Average Response Time (Seconds)
2/2/2004	3.0
2/3/2004	3.2
2/4/2004	2.1
2/5/2004	1.1
2/6/2004	4.0
Average over period	2.7

3.1.7 Execution Load by Agent Group For Transaction <transaction>



**Execution Load by Agent Group
For Transaction <transaction>**

Agent Group Name

Chicago Servers

Date	Transaction Count
2/2/2004	2043
2/3/2004	2165
2/4/2004	2672
2/5/2004	2982
2/6/2004	4361
Total for period	14223
Average over period	2844.6

Agent Group Name

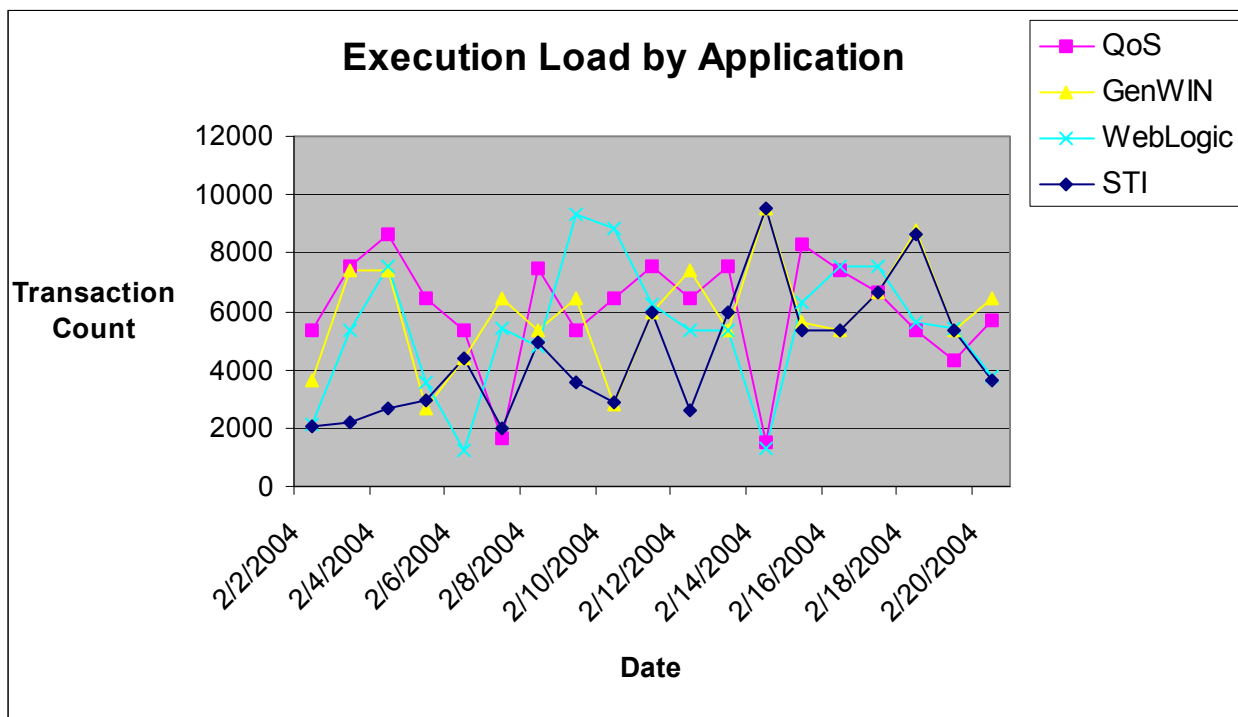
Milwaukee Servers

Date	Transaction Count
2/2/2004	5343
2/3/2004	7545
2/4/2004	8652
2/5/2004	6453
2/6/2004	5361
Total for period	33354
Average over period	6670.8

Agent Group Name	
Minneapolis Servers	
Date	Transaction Count
2/2/2004	3641
2/3/2004	7435
2/4/2004	7432
2/5/2004	2642
2/6/2004	4361
Total for period	25511
Average over period	5102.2

Agent Group Name	
Detroit Servers	
Date	Transaction Count
2/2/2004	2099
2/3/2004	5365
2/4/2004	7509
2/5/2004	3582
2/6/2004	1261
Total for period	19816
Average over period	3963.2

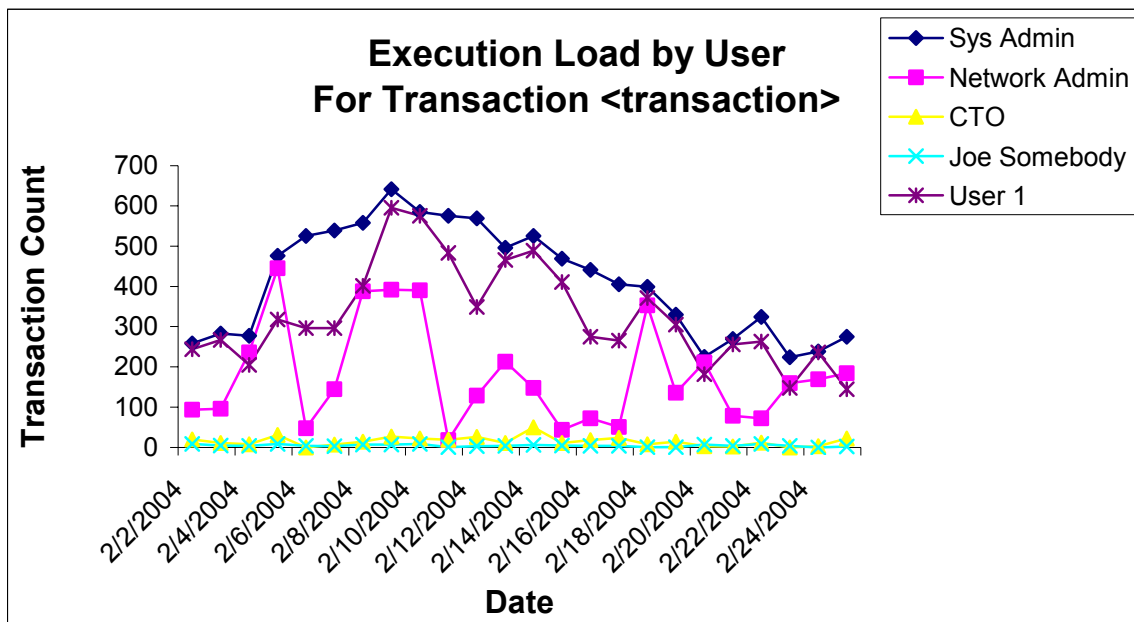
3.1.8 Execution Load by Application



Execution Load by Application

Application Name	
STI	
Date	Transaction Count
2/2/2004	2043
2/3/2004	2165
2/4/2004	2672
2/5/2004	2982
2/6/2004	4361
Total for period	14223
Average over period	2844.6
Application Name	
QoS	
Date	Transaction Count
2/2/2004	5343
2/3/2004	7545
2/4/2004	8652
2/5/2004	6453
2/6/2004	5361
Total for period	33354
Average over period	6670.8
Application Name	
GenWIN	
Date	Transaction Count
2/2/2004	3641
2/3/2004	7435
2/4/2004	7432
2/5/2004	2642
2/6/2004	4361
Total for period	25511
Average over period	5102.2
Application Name	
WebLogic	
Date	Transaction Count
2/2/2004	2099
2/3/2004	5365
2/4/2004	7509
2/5/2004	3582
2/6/2004	1261
Total for period	19816
Average over period	3963.2

3.1.9 Execution Load by User For Transaction <transaction>



Execution Load by User
For Transaction <transaction>

User Name

Sys Admin

Date	Transaction Count
2/2/2004	258
2/3/2004	283
2/4/2004	277
2/5/2004	476
2/6/2004	525
Total for period	1819
Average over period	363.8

User Name

Network Admin

Date	Transaction Count
2/2/2004	117
2/3/2004	126
2/4/2004	143
2/5/2004	211
2/6/2004	45
Total for period	642
Average over period	128.4

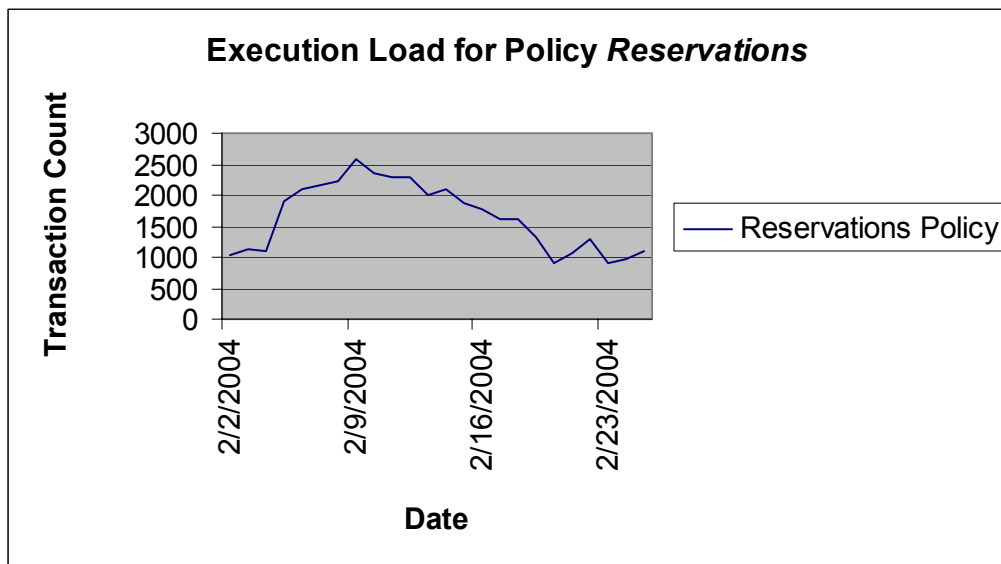
User Name

CTO

Date	Transaction Count
2/2/2004	10
2/3/2004	18

2/4/2004	10
2/5/2004	46
2/6/2004	8
Total for period	92
Average over period	18.4
<hr/>	
User Name	
Joe Somebody	
<hr/>	
Date	Transaction Count
2/2/2004	0
2/3/2004	3
2/4/2004	9
2/5/2004	6
2/6/2004	2
Total for period	20
Average over period	4
<hr/>	
User Name	
User 1	
<hr/>	
Date	Transaction Count
2/2/2004	129
2/3/2004	177
2/4/2004	202
2/5/2004	276
2/6/2004	275
Total for period	1059
Average over period	211.8

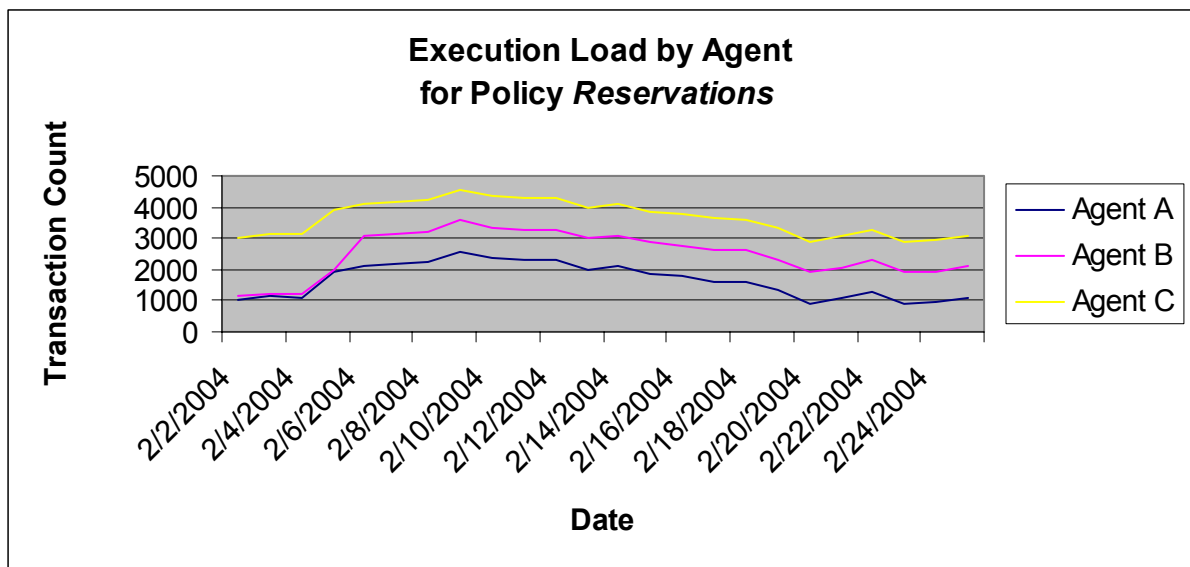
3.1.10 Execution Load For Policy *Reservations*



Execution Load for Policy *Reservations*

Date	Transaction Count
2/2/2004	1032
2/3/2004	1132
2/4/2004	1109
2/5/2004	1905
2/6/2004	2103
Total for period	7281
Average for period	1456.2

3.1.11 Execution Load by Agent for Policy *Policy*



Execution Load by Agent For Policy Reservations

Agent Name

Agent A

Date	Transaction Count
2/2/2004	1032
2/3/2004	1132
2/4/2004	1109
2/5/2004	1905
2/6/2004	2103
Total over period	7281
Average over period	1456.2

Agent Name

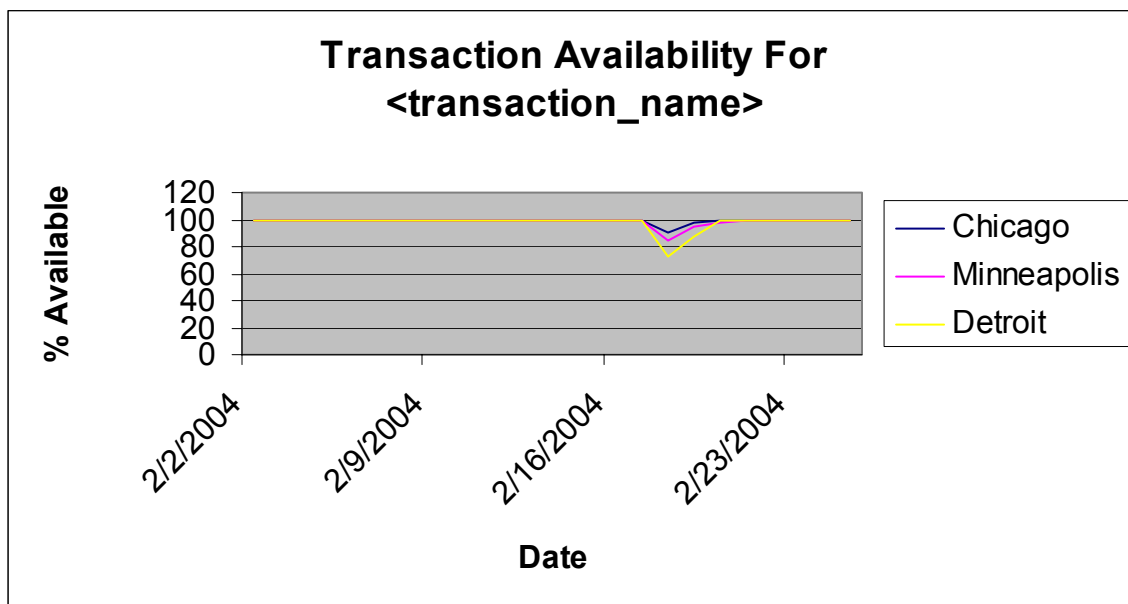
Agent B

Date	Transaction Count
2/2/2004	1132
2/3/2004	1232
2/4/2004	1209

2/5/2004	2905
2/6/2004	3103
Total over period	9581
Average over period	1916.2
Agent Name	
Agent C	

Date	Transaction Count
2/2/2004	3032
2/3/2004	3132
2/4/2004	3109
2/5/2004	3905
2/6/2004	4103
Total over period	17281
Average over period	3456.2

3.1.12 Transaction Availability For <transaction>



Agent Name

Chicago

Date	% Available
2/2/2004	100%
2/3/2004	100%
2/4/2004	100%
2/5/2004	100%
2/6/2004	100%
Average availability over period	100%

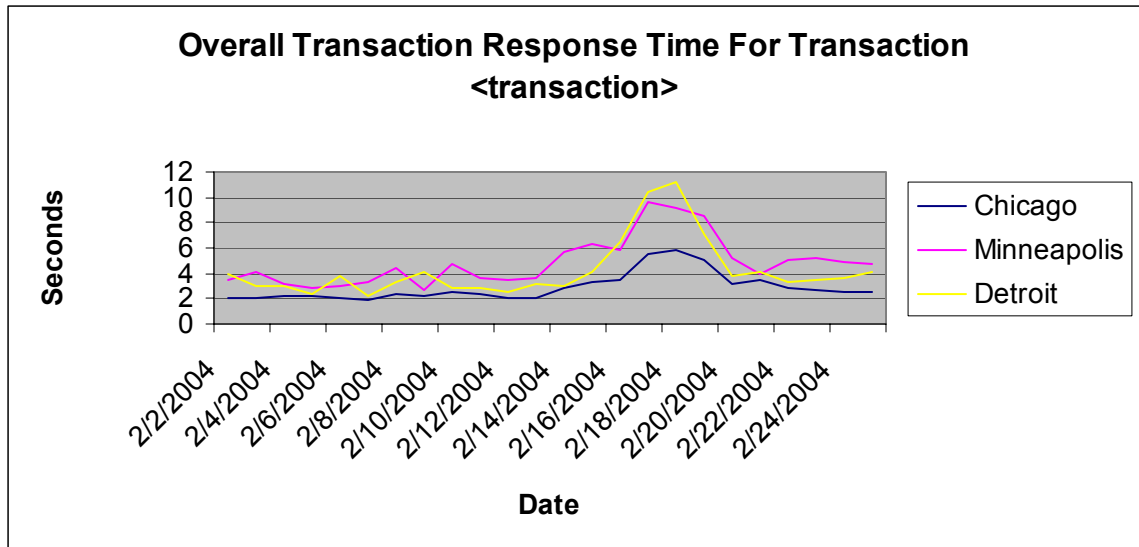
Agent Name**Detroit**

Date	% Available
2/2/2004	100%
2/3/2004	100%
2/4/2004	100%
2/5/2004	100%
2/6/2004	100%
Average availability over period	100%

Agent Name**Minneapolis**

Date	% Available
2/2/2004	100%
2/3/2004	100%
2/4/2004	100%
2/5/2004	100%
2/6/2004	100%
Average availability over period	100%

3.1.13 Overall Transaction Response Time For Transaction <transaction>



Overall Transaction Response For Transaction <transaction>

Agent Name**Chicago**

Date	Average Response Time (Seconds)
2/2/2004	2.1
2/3/2004	1.9

2/4/2004	2.0
2/5/2004	2.2
2/6/2004	2.1
Average over period	2.1

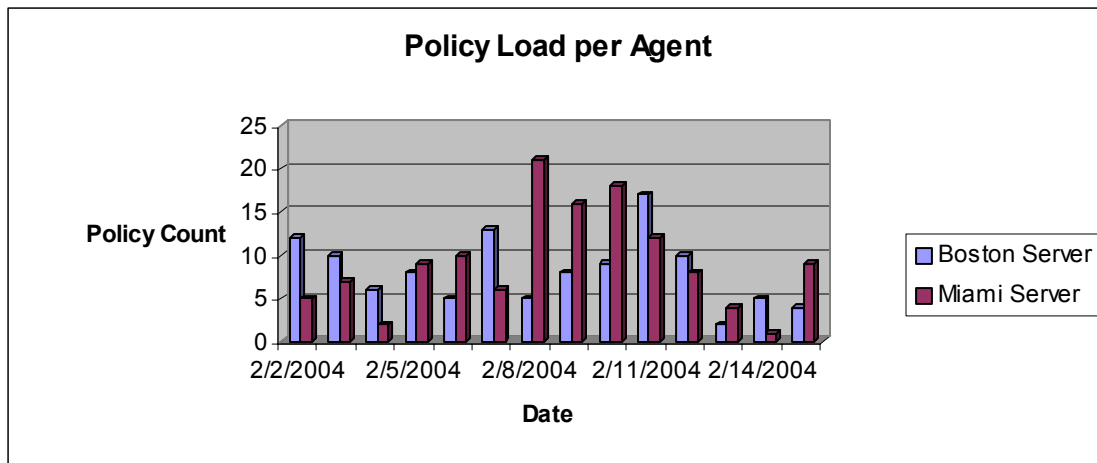
Agent Name	
Detroit	
Date	Average Response Time (Seconds)
2/2/2004	4.0
2/3/2004	2.2
2/4/2004	4.1
2/5/2004	2.1
2/6/2004	3.0
Average over days shown	3.1

Agent Name	
Minneapolis	
Date	Average Response Time (Seconds)
2/2/2004	3.1
2/3/2004	2.0
2/4/2004	3.9
2/5/2004	3.7
2/6/2004	3.5
Average over days shown	3.3

3.1.14 Slowest Transactions

SLOWEST TRANSACTIONS							
<u>Application</u>	<u>Transaction</u>	<u>User</u>	<u>Agent</u>	<u>Average Response Time (Seconds)</u>	<u>Minimum Response Time (Seconds)</u>	<u>Maximum Response Time (Seconds)</u>	<u>Transaction Start Date</u>
WAS	TransactionA	NetPerf Admin	Chicago Server	5.6	0.75	15.6	2004-02-02
STI	TransactionB	App Admin	Chicago Server	5.1	0.99	10.1	2004-02-03

3.1.15 Policy Load per Agent

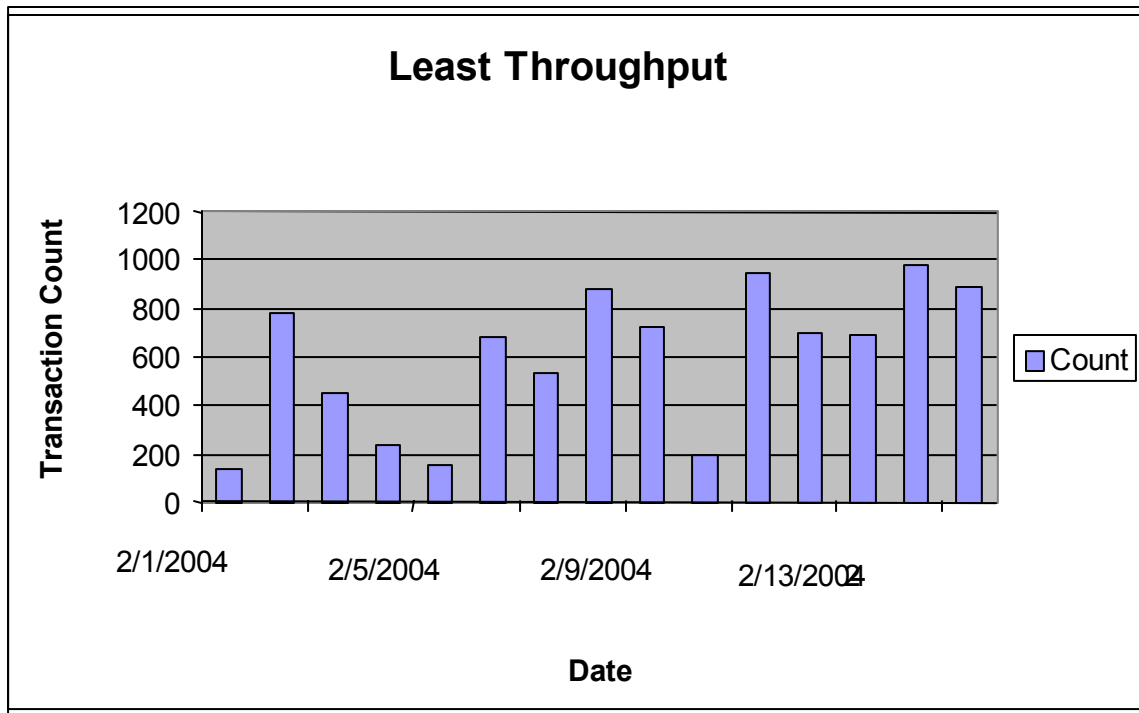
**Policy Load per Agent****Agent Name****Boston Server**

<u>Date</u>	<u>Policy Count</u>
2/2/2004	12
2/3/2004	10
2/4/2004	6
2/5/2004	8
2/6/2004	5
Total for period	41
Average over period	8.2

Agent Name**Miami Server**

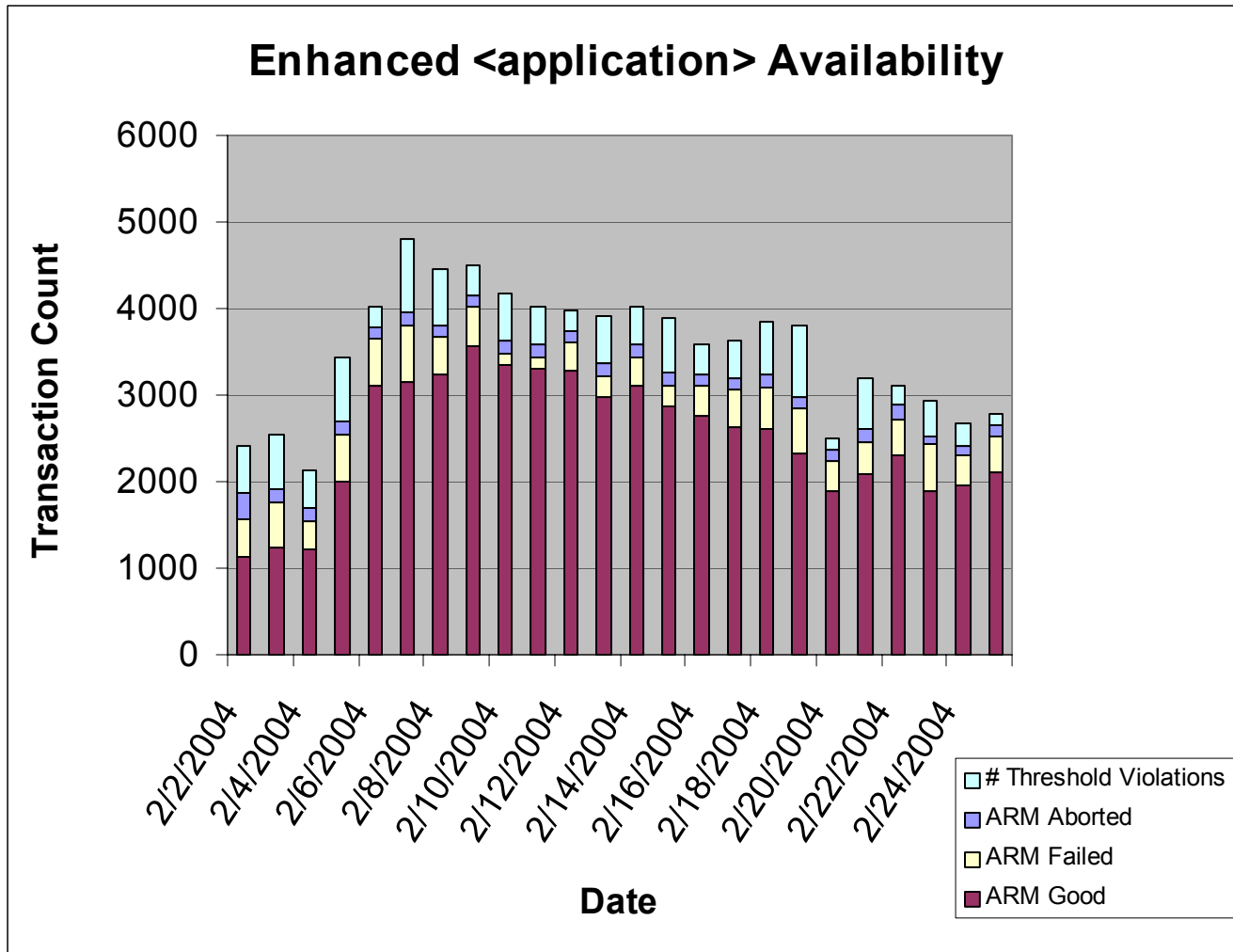
<u>Date</u>	<u>Policy Count</u>
2/2/2004	5
2/3/2004	7
2/4/2004	2
2/5/2004	9
2/6/2004	10
Total for period	33
Average over period	6.6

3.1.16 Least Throughput



Least Throughput					
<u>Application</u>	<u>Transaction Name</u>	<u>User</u>	<u>Agent</u>	<u>Transaction Count</u>	<u>Transaction Start Date</u>
WAS	TransactionA	NetPerf Admin	Chicago Server	350	2004-02-02
STI	TransactionB	App Admin	Chicago Server	560	2004-02-03

3.1.17 Enhanced <application> Availability



Enhanced Application Availability

Application Name - ARM

Date

2/2/2004	Total Transaction Count	1874
	Transaction Count - Successful	1132
	Transaction Count - Failed	430
	Transaction Count - Aborted	312
	Transaction Count - Threshold Violations	530

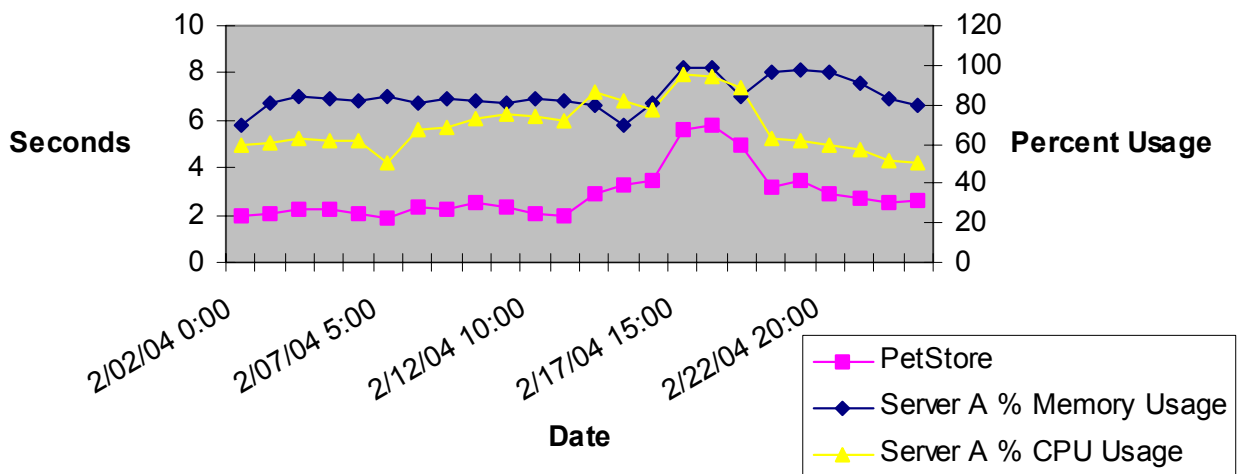
Date

2/3/2004	Total Transaction Count	1905
	Transaction Count - Successful	1232
	Transaction Count - Failed	532
	Transaction Count - Aborted	141
	Transaction Count - Threshold Violations	632

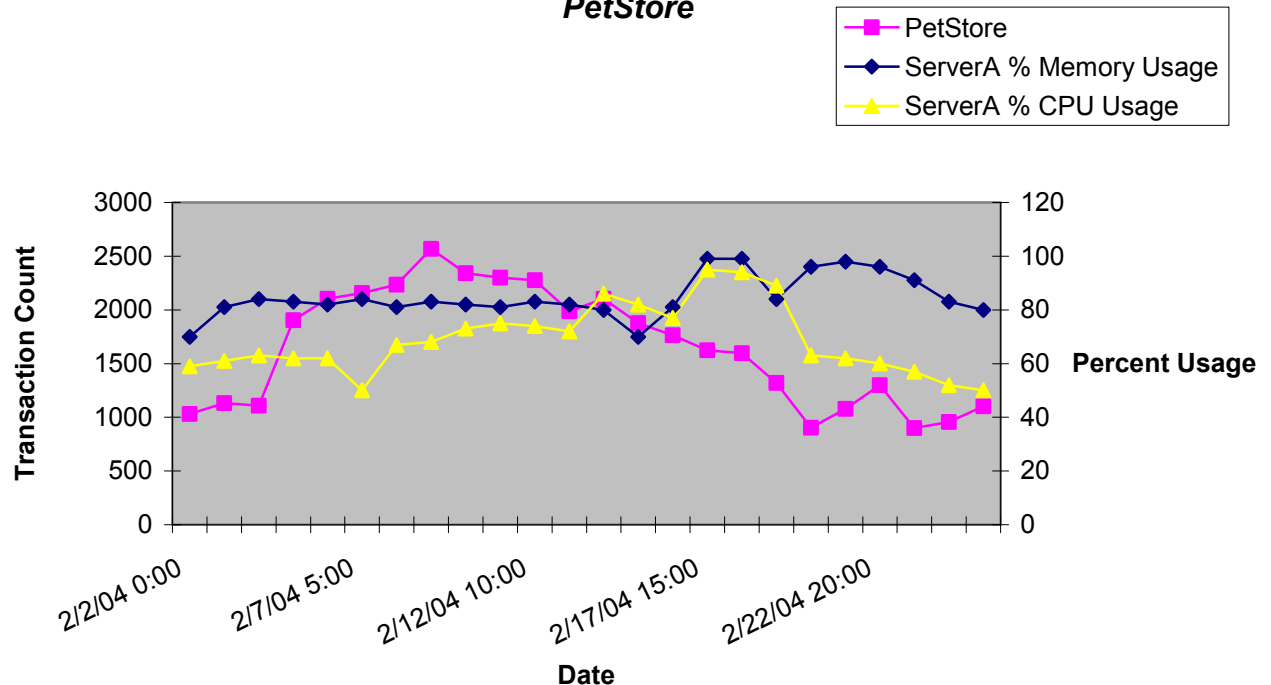
3.1.18 Resource Usage and Transaction Response Time for Transaction *Petstore*

Cross Application Report using Tivoli Monitoring for Transaction Performance and IBM Tivoli Monitoring warehouse data

Resource Usage and Response Time For Transaction *PetStore*



Resource Usage and Transaction Execution Load for Transaction *PetStore*



Resource Usage, Response Time and Execution Load for Transaction *PetStore*

<u>Date</u>	<u>Average Response Time (Seconds)</u>	<u>% Memory Usage</u>	<u>% CPU Usage</u>	<u>Transaction Count</u>
2004-02-02	2.3	73	88	560
2004-02-03	1.9	83	84	350
2004-02-04	--	75	89	707
<u>Overall Average</u>	2.1	78	86	455

3.2 Report Parameter Inputs Defined

Each report has a set of input parameters that should be selected before running the report. For each parameter enter a value and click **Select** so the selection will appear in the box below of choices for that parameter. Any parameters typed in should be entered using the same case and spacing as the data was stored in the database with. After all parameters have been entered and selected, click **OK** at the bottom of the parameter page.

An '*' value means to return all possible values for that parameter. The icon just to the right of the Start Date and End Date input field will pop up a calendar that you point and click on to select the desired dates. When selecting how much data to show in a report consider the space available for the graph in the web browser and limit the Start Date and End Date fields appropriately for optimal graphic display

capabilities. It is recommended to only show up 30 data points at a time as excessive amounts of data to plot in one graph may render the graph difficult to read.

Some input parameters have a 'Set to Null' box associated with an input parameter. These should not be used and will be removed in future releases.

3.2.1 Average Response Time by Application

For this report, the average response time trend is shown over the time period requested and then grouped by application. The average response time is calculated as shown in the following calculation:

$$\text{sum}(\text{avg_response_time} * \text{transaction_execution_count}) / \text{sum}(\text{transaction_execution_count})$$

3.2.1.1 Start Date and End Date

The Start_Date and End_Date input parameter values are chosen from a pop-up calendar before running the report so all data retrieved is greater than or equal to the Start_Date value and less than or equal to the End_Date value. Only one value is permitted for each parameter. When the WEEK time granularity is chosen the data is dated for the beginning Sunday of each week and this date must fall on or between the Start and End dates to be shown. When the MONTH time granularity is chosen the data is dated for the first day of each month and this date must fall on or between the Start and End dates to be shown.

3.2.1.2 Time Granularity

The time granularity input parameter is a drop-down menu where you can select from one of HOUR, DAY, WEEK, or MONTH. This parameter is used to select from which fact table to read.

3.2.1.3 Application Name

The application name input parameter can be a single application name or can be a wild card. To get all applications use the '*' wild card value or to get a set of applications use wild carding like AC* or *AC*. This input parameter is a text field and only characters should be entered for the application name.

3.2.2 Average Response Time by User for Transaction <transaction>

For this report, the average response time trend is shown over the time period requested and for the transaction specified; it is then grouped by the invoking user, showing only edge transactions. The edge transactions will have a value of 'N/A' in the BWMD_TX_ND table. This information will be useful when choosing which transaction names to view for this report. The average response time is calculated as shown in the following calculation:

$$\text{sum}(\text{avg_response_time} * \text{transaction_execution_count}) / \text{sum}(\text{transaction_execution_count})$$

3.2.2.1 Start Date and End Date

The Start_Date and End_Date input parameter values are chosen from a pop-up calendar before running the report so all data retrieved is greater than or equal to the Start_Date value and less than or equal to the End_Date value. Only one value is permitted for each parameter. When the WEEK time granularity is chosen the data is dated for the beginning Sunday of each week and this date must fall on or between the Start and End dates to be shown. When the MONTH time granularity is chosen the data is dated for the first day of each month and this date must fall on or between the Start and End dates to be shown.

3.2.2.2 Time Granularity

The time granularity input parameter is a drop-down menu where you can select from HOUR, DAY, WEEK, or MONTH. This parameter is used to select from which fact table the data will be read.

3.2.2.3 User Name

The user name input parameter can be a single, specific user name or can be a wild card. To get all user names use the * wild card value or, to get a set of user names, use wild carding like AC* or *AC*. If there are a large number of users to show on the report the graph may not be able to scale to show all the user names, so only a reasonable quantity should be selected for viewing.

3.2.2.4 Transaction Name

The transaction name input parameter should be entered as a single, exact value. The graph shows average response time for the transaction and users selected.

3.2.3 Average Response Time by Policy

For this report, the average response time trend is shown over the time period requested and then grouped by management policy. The average response time is calculated as shown in the following calculation:

$$\text{sum}(\text{avg_response_time} * \text{transaction_execution_count}) / \text{sum}(\text{transaction_execution_count})$$

This translates to the following fact table columns:

$$\text{sum}(\text{fact_table.sample_count} * \text{fact_table.sample_count}) / \text{sum}(\text{fact_table.sample_count})$$

3.2.3.1 Start Date and End Date

The Start_Date and End_Date input parameter values are chosen from a pop-up calendar before running the report so all data retrieved is greater than or equal to the Start_Date value and less than or equal to the End_Date value. Only one value is permitted for each parameter. When the WEEK time granularity is chosen the data is dated for the beginning Sunday of each week and this date must fall on or between the Start and End dates to be shown. When the MONTH time granularity is chosen the data is dated for the first day of each month and this date must fall on or between the Start and End dates to be shown.

3.2.3.2 Time Granularity

The time granularity input parameter is a drop-down menu where you can select from HOUR, DAY, WEEK, or MONTH. This parameter is used to select from which fact table the data will be read.

3.2.3.3 Policy Name

The policy name input parameter can be a single, specific policy name or can a wild card. To get all policy names use the '*' wild card value or, to get a set of policy names, use wild carding like AC* or *AC*. If there are a large number of policies to show on the report the graph may not be able to scale to show all the policy names, so only a reasonable quantity should be selected for viewing.

3.2.4 Average Response Time by Policy Group

For this report, the average response time trend is shown over the time period requested and then grouped by management policy group. The average response time is calculated as shown in the following calculation:

$$\text{sum}(\text{avg_response_time} * \text{transaction_execution_count}) / \text{sum}(\text{transaction_execution_count})$$

This translates to the following fact table columns:

$$\text{sum}(\text{fact_table.sample_count} * \text{fact_table.sample_count}) / \text{sum}(\text{fact_table.sample_count})$$

3.2.4.1 Start Date and End Date

The Start_Date and End_Date input parameter values are chosen from a pop-up calendar before running the report so all data retrieved is greater than or equal to the Start_Date value and less than or equal to the

End_Date value. Only one value is permitted for each parameter. When the WEEK time granularity is chosen the data is dated for the beginning Sunday of each week and this date must fall on or between the Start and End dates to be shown. When the MONTH time granularity is chosen the data is dated for the first day of each month and this date must fall on or between the Start and End dates to be shown.

3.2.4.2 Time Granularity

The time granularity input parameter is a drop-down menu where you can select from one of HOUR, DAY, WEEK, or MONTH. This parameter is used to select from which fact table the data will be read.

3.2.4.3 Policy Group Name

The policy group name input parameter can be a single, specific policy group name or can be wild carded. To get all policy group names use the '*' wild card value or, to get a set of policy group names, use wild carding like AC* or *AC*. If there are a large number of policy groups to show on the report the graph may not be able to scale to show all the policy group names, so only a reasonable quantity should be selected for viewing.

3.2.5 Average Response Time by Policy For Policy Group <policy_group>

For this report, the average response time trend is shown over the time period requested and the policy group specified and then grouped by management policy. The average response time is calculated as shown in the following calculation:

$$\text{sum}(\text{avg_response_time} * \text{transaction_execution_count}) / \text{sum}(\text{transaction_execution_count})$$

This translates to the following fact table columns:

$$\text{sum}(\text{fact_table.sample_count} * \text{fact_table.sample_count}) / \text{sum}(\text{fact_table.sample_count})$$

3.2.5.1 Start Date and End Date

The Start_Date and End_Date input parameter values are chosen from a pop-up calendar before running the report so all data retrieved is greater than or equal to the Start_Date value and less than or equal to the End_Date value. Only one value is permitted for each parameter. When the WEEK time granularity is chosen the data is dated for the beginning Sunday of each week and this date must fall on or between the Start and End dates to be shown. When the MONTH time granularity is chosen the data is dated for the first day of each month and this date must fall on or between the Start and End dates to be shown.

3.2.5.2 Time Granularity

The time granularity input parameter is a drop-down menu where you can select from one of HOUR, DAY, WEEK, or MONTH. This parameter is used to select which fact table the data will be read from.

3.2.5.3 Policy Group Name

The policy group name input parameter should be entered as a single, exact value. The graph shows the average response time for each policy belonging to the policy group selected.

3.2.6 Average Response Time by Agent Group for Transaction <transaction>

For this report, the average response time trend is shown over the time period requested and for the transaction specified and then grouped by agent group. The average response time is calculated as shown in the following calculation:

$$\text{sum}(\text{avg_response_time} * \text{transaction_execution_count}) / \text{sum}(\text{transaction_execution_count})$$

This translates to the following fact table columns:

$$\text{sum}(\text{fact_table.sample_count} * \text{fact_table.sample_count}) / \text{sum}(\text{fact_table.sample_count})$$

3.2.6.1 Start Date and End Date

The Start_Date and End_Date input parameter values are chosen from a pop-up calendar before running the report so all data retrieved is greater than or equal to the Start_Date value and less than or equal to the End_Date value. Only one value is permitted for each parameter. When the WEEK time granularity is chosen the data is dated for the beginning Sunday of each week and this date must fall on or between the Start and End dates to be shown. When the MONTH time granularity is chosen the data is dated for the first day of each month and this date must fall on or between the Start and End dates to be shown.

3.2.6.2 Time Granularity

The time granularity input parameter is a drop-down menu where you can select from one of HOUR, DAY, WEEK, or MONTH. This parameter is used to select from which fact table the data will be read.

3.2.6.3 Agent Group Name

The agent group name input parameter can be a single, specific agent group name or can be wild carded. To get all agent group names use the '*' wild card value or, to get a set of agent group names, use wild carding like AC* or *AC*. If there are a large number of agent groups to show on the report the graph may not be able to scale to show all the agent group names, so only a reasonable quantity should be selected for viewing.

3.2.6.4 Transaction Name

The transaction name input parameter should be entered as a single, exact value. The graph shows average response time for the transaction and agent groups selected.

3.2.7 Execution Load by Agent Group For Transaction <transaction>

For this report, the transaction execution count trend is shown over the time period requested and for the transaction specified and then grouped by agent group. The transaction execution count is calculated as shown in the following calculation:

$$\text{sum}(\text{transaction_execution_count})$$

This translates to the following fact table columns:

$$\text{sum}(\text{fact_table.sample_count})$$

3.2.7.1 Start Date and End Date

The Start_Date and End_Date input parameter values are chosen from a pop-up calendar before running the report so all data retrieved is greater than or equal to the Start_Date value and less than or equal to the End_Date value. Only one value is permitted for each parameter. When the WEEK time granularity is chosen the data is dated for the beginning Sunday of each week and this date must fall on or between the Start and End dates to be shown. When the MONTH time granularity is chosen the data is dated for the first day of each month and this date must fall on or between the Start and End dates to be shown.

3.2.7.2 Time Granularity

The time granularity input parameter is a drop-down menu where you can select from one of HOUR, DAY, WEEK, or MONTH. This parameter is used to select from which fact table the data will be read.

3.2.7.3 Agent Group Name

The agent group name input parameter can be a single, specific agent group name or can be wild carded. To get all agent group names use the '*' wild card value or, to get a set of agent group names, use wild carding like AC* or *AC*. If there are a large number of agent groups to show on the report the graph may not be able to scale to show all the agent group names, so only a reasonable quantity should be selected for viewing.

3.2.7.4 Transaction Name

The transaction name input parameter should be entered as a single, exact value. The graph shows execution load for the transaction and agent groups selected.

3.2.8 Execution Load by Application

For this report, the transaction execution count trend is shown over the time period requested and then grouped by application. The transaction execution count is calculated as shown in the following calculation:

```
sum(transaction_execution_count)
```

This translates to the following fact table columns:

```
sum(fact_table.sample_count)
```

3.2.8.1 Start Date and End Date

The Start_Date and End_Date input parameter values are chosen from a pop-up calendar before running the report so all data retrieved is greater than or equal to the Start_Date value and less than or equal to the End_Date value. Only one value is permitted for each parameter. When the WEEK time granularity is chosen the data is dated for the beginning Sunday of each week and this date must fall on or between the Start and End dates to be shown. When the MONTH time granularity is chosen the data is dated for the first day of each month and this date must fall on or between the Start and End dates to be shown.

3.2.8.2 Time Granularity

The time granularity input parameter is a drop-down menu where you can select from one of HOUR, DAY, WEEK, or MONTH. This parameter is used to select from which fact table the data will be read. .

3.2.8.3 Application Name

The application name input parameter can be a single application name or can be wild carded. To get all applications use the '*' wild card value or, to get a set of applications, use wild carding like AC* or *AC*. This input parameter is a text field and only characters should be entered for the application name.

3.2.9 Execution Load by User For Transaction <transaction>

For this report, the transaction execution count trend is shown over the time period requested and for the transaction specified and then grouped by the invoking user, showing only edge transactions. The edge transactions will have a value of 'N/A' in the BWM.D_TX_ND table. This information will be useful when choosing which transaction names to view for this report. The transaction execution count is calculated as shown in the following calculation:

```
sum(transaction_execution_count)
```

This translates to the following fact table columns:

```
sum(fact_table.sample_count)
```

3.2.9.1 Start Date and End Date

The Start_Date and End_Date input parameter values are chosen from a pop-up calendar before running the report so all data retrieved is greater than or equal to the Start_Date value and less than or equal to the End_Date value. Only one value is permitted for each parameter. When the WEEK time granularity is chosen the data is dated for the beginning Sunday of each week and this date must fall on or between the Start and End dates to be shown. When the MONTH time granularity is chosen the data is dated for the first day of each month and this date must fall on or between the Start and End dates to be shown.

3.2.9.2 Time Granularity

The time granularity input parameter is a drop-down menu where you can select from one of HOUR, DAY, WEEK, or MONTH. This parameter is used to select from which fact table the data will be read.

3.2.9.3 User Name

The user name input parameter can be a single, specific user name or can be wild carded. To get all user names use the '*' wild card value or, to get a set of user names, use wild carding like AC* or *AC*. If there are a large number of users to show on the report the graph may not be able to scale to show all the user names, so only a reasonable quantity should be selected for viewing.

3.2.9.4 Transaction Name

The transaction name input parameter should be entered as a single, exact value. The graph shows execution load for the transaction and users selected.

3.2.10 Execution Load For Policy <policy>

For this report, the transaction execution count trend is shown over the time period requested and then grouped by management policy. The transaction execution count is calculated as shown in the following calculation:

```
sum(transaction_execution_count)
```

This translates to the following fact table columns:

```
sum(fact_table.sample_count)
```

3.2.10.1 Start Date and End Date

The Start_Date and End_Date input parameter values are chosen from a pop-up calendar before running the report so all data retrieved is greater than or equal to the Start_Date value and less than or equal to the End_Date value. Only one value is permitted for each parameter. When the WEEK time granularity is chosen the data is dated for the beginning Sunday of each week and this date must fall on or between the Start and End dates to be shown. When the MONTH time granularity is chosen the data is dated for the first day of each month and this date must fall on or between the Start and End dates to be shown.

3.2.10.2 Time Granularity

The time granularity input parameter is a drop-down menu where you can select from one of HOUR, DAY, WEEK, or MONTH. This parameter is used to select from which fact table the data will be read. .

3.2.10.3 Policy Name

The policy name input parameter can be a single, specific policy name or can be wild carded. To get all policy names use the '*' wild card value or, to get a set of policy names, use wild carding like AC* or *AC*. If there are a large number of policies to show on the report the graph may not be able to scale to show all the policy names, so only a reasonable quantity should be selected for viewing.

3.2.11 Execution Load by Agent For Policy <policy>

For this report, the transaction execution count trend is shown over the time period requested and for the management policy specified and then grouped by hosts found for that management policy. The transaction execution count is calculated as shown in the following calculation:

$\text{sum}(\text{transaction_execution_count})$

This translates to the following fact table columns:

$\text{sum}(\text{fact_table.sample_count})$

3.2.11.1 Start Date and End Date

The Start_Date and End_Date input parameter values are chosen from a pop-up calendar before running the report so all data retrieved is greater than or equal to the Start_Date value and less than or equal to the End_Date value. Only one value is permitted for each parameter. When the WEEK time granularity is chosen the data is dated for the beginning Sunday of each week and this date must fall on or between the Start and End dates to be shown. When the MONTH time granularity is chosen the data is dated for the first day of each month and this date must fall on or between the Start and End dates to be shown.

3.2.11.2 Time Granularity

The time granularity input parameter is a drop-down menu where you can select from one of HOUR, DAY, WEEK, or MONTH. This parameter is used to select from which fact table the data will be read.

3.2.11.3 Policy Name

The policy name input parameter can be a single, specific policy name or can be wild carded. To get all policy names use the '*' wild card value or, to get a set of policy names, use wild carding like AC* or *AC*. If there are a large number of policies to show on the report the graph may not be able to scale to show all the policy names, so only a reasonable quantity should be selected for viewing.

3.2.12 Transaction Availability For Transaction <transaction>

For this report, the percent of transaction availability trend is shown over the time period requested and then grouped by transaction name, host name and date. The transaction availability is calculated as shown in the following calculation:

$\text{sum}(\text{transaction_execution_count}) - \text{sum}(\text{transaction execution error count})$

 $\text{sum}(\text{transaction_execution_count})$

This translates to the following fact table columns:

$\text{sum}(\text{fact_table.sample_count}) - \text{sum}(\text{fact_table.error_count})$

 $\text{sum}(\text{fact_table.sample_count})$

3.2.12.1 Start Date and End Date

The Start_Date and End_Date input parameter values are chosen from a pop-up calendar before running the report so all data retrieved is greater than or equal to the Start_Date value and less than or equal to the End_Date value. Only one value is permitted for each parameter. When the WEEK time granularity is chosen the data is dated for the beginning Sunday of each week and this date must fall on or between the

Start and End dates to be shown. When the MONTH time granularity is chosen the data is dated for the first day of each month and this date must fall on or between the Start and End dates to be shown.

3.2.12.2 Time Granularity

The time granularity input parameter is a drop-down menu where you can select from one of HOUR, DAY, WEEK, or MONTH. This parameter is used to select from which fact table the data will be read.

3.2.12.3 Agent Name

The agent name, or hostname, input parameter can be a single, specific policy name or can be wild carded. To get all policy names use the '*' wild card value or, to get a set of policy names, use wild carding like AC* or *AC*. If there are a large number of agents to show on the report the graph may not be able to scale to show all the agent names, so only a reasonable quantity should be selected for viewing.

3.2.12.4 Transaction Name

The transaction name input parameter can be a single, specific transaction name or can be wild carded. To get all transaction names use the '*' wild card value or, to get a set of transaction names, use wild carding like AC* or *AC*. If there are a large number of transactions to show on the report the graph may not be able to scale to show all the transaction names, so only a reasonable quantity should be selected for viewing.

3.2.13 Overall Transaction Response Time For Transaction <transaction>

For this report, the transaction average response time trend is shown over the time period requested and then grouped by transaction name, host name and date. The average response time is calculated as shown in the following calculation:

$$\text{sum}(\text{avg_response_time} * \text{transaction_execution_count}) / \text{sum}(\text{transaction_execution_count})$$

This translates to the following fact table columns:

$$\text{sum}(\text{fact_table.sample_count} * \text{fact_table.sample_count}) / \text{sum}(\text{fact_table.sample_count})$$

3.2.13.1 Start Date and End Date

The Start_Date and End_Date input parameter values are chosen from a pop-up calendar before running the report so all data retrieved is greater than or equal to the Start_Date value and less than or equal to the End_Date value. Only one value is permitted for each parameter. When the WEEK time granularity is chosen the data is dated for the beginning Sunday of each week and this date must fall on or between the Start and End dates to be shown. When the MONTH time granularity is chosen the data is dated for the first day of each month and this date must fall on or between the Start and End dates to be shown.

3.2.13.2 Time Granularity

The time granularity input parameter is a drop-down menu where you can select from one of HOUR, DAY, WEEK, or MONTH. This parameter is used to select from which fact table the data will be read.

3.2.13.3 Agent Name

The agent name, or hostname, input parameter can be a single, specific policy name or can be wild carded. To get all policy names use the '*' wild card value or, to get a set of policy names, use wild carding like AC* or *AC*. If there are a large number of agents to show on the report the graph may not be able to scale to show all the agent names, so only a reasonable quantity should be selected for viewing.

3.2.13.4 Transaction Name

The transaction name input parameter can be a single, specific transaction name or can be wild carded. To get all transaction names use the '*' wild card value or, to get a set of transaction names, use wild carding like AC* or *AC*. If there are a large number of transactions to show on the report the graph may not be able to scale to show all the transaction names, so only a reasonable quantity should be selected for viewing.

3.2.14 Policy Load per Agent

For this report, the policy load per agent count trend is shown over the time period requested and then grouped by host name. The policy load count is calculated as shown in the following calculation:

```
count(distinct management_policy)
```

This translates to the following D_TX_ND dimension table column:

```
count(distinct management_policy)
```

3.2.14.1 Start Date and End Date

The Start_Date and End_Date input parameter values are chosen from a pop-up calendar before running the report so all data retrieved is greater than or equal to the Start_Date value and less than or equal to the End_Date value. Only one value is permitted for each parameter. When the WEEK time granularity is chosen the data is dated for the beginning Sunday of each week and this date must fall on or between the Start and End dates to be shown. When the MONTH time granularity is chosen the data is dated for the first day of each month and this date must fall on or between the Start and End dates to be shown.

3.2.14.2 Time Granularity

The time granularity input parameter is a drop-down menu where you can select from one of HOUR, DAY, WEEK, or MONTH. This parameter is used to select from which fact table the data will be read.

3.2.14.3 Agent Name

The agent name, or hostname, input parameter can be a single, specific policy name or can be wild carded. To get all policy names use the '*' wild card value or, to get a set of policy names, use wild carding like AC* or *AC*. If there are a large number of agents to show on the report the graph may not be able to scale to show all the agent names, so only a reasonable quantity should be selected for viewing.

3.2.15 Least Throughput

For this report, the transaction run count trend is shown over the time period requested. The detailed report shows the transaction count numbers in ascending order for each unique combination of transaction, user, agent and application. This shows the transactions that were executed the least first. The transaction execution count is calculated as shown in the following calculation:

```
sum(transaction_execution_count)
```

This translates to the following fact table columns:

```
sum(fact_table.sample_count)
```

3.2.15.1 Start Date and End Date

The Start_Date and End_Date input parameter values are chosen from a pop-up calendar before running the report so all data retrieved is greater than or equal to the Start_Date value and less than or equal to the End_Date value. Only one value is permitted for each parameter. When the WEEK time granularity is chosen the data is dated for the beginning Sunday of each week and this date must fall on or between the

Start and End dates to be shown. When the MONTH time granularity is chosen the data is dated for the first day of each month and this date must fall on or between the Start and End dates to be shown.

3.2.15.2 Time Granularity

The time granularity input parameter is a drop-down menu where you can select from one of HOUR, DAY, WEEK, or MONTH. This parameter is used to select from which fact table the data will be read.

3.2.15.3 Agent Name

The agent name, or hostname, input parameter can be a single, specific policy name or can be wild carded. To get all policy names use the '*' wild card value or, to get a set of policy names, use wild carding like AC* or *AC*. If there are a large number of agents to show on the report the graph may not be able to scale to show all the agent names, so only a reasonable quantity should be selected for viewing.

3.2.15.4 Application Name

The application name input parameter can be a single application name or can be wild carded. To get all applications use the '*' wild card value or to get a set of applications use wild carding like AC* or *AC*. This input parameter is a text field and only characters should be entered for the application name.

3.2.15.5 User Name

The user name input parameter can be a single, specific user name or can be wild carded. To get all user names use the '*' wild card value or, to get a set of user names, use wild carding AC* or *AC*. If there are a large number of users to show on the report the graph may not be able to scale to show all the user names, so only a reasonable quantity should be selected for viewing.

3.2.15.6 Transaction Name

The transaction name input parameter can be a single, specific transaction name or can be wild carded. To get all transaction names use the '*' wild card value or, to get a set of transaction names, use wild carding like AC* or *AC*. If there are a large number of transactions to show on the report the graph may not be able to scale to show all the transaction names, so only a reasonable quantity should be selected for viewing.

3.2.15.7 Number of Transactions

The 'Maximum Number of Transactions' input parameter defines the limit of how many transactions show in the report. The transactions with the lowest run count will be shown in ascending order.

3.2.16 Slowest Transactions

For this report, the transaction average response time trend is shown over the time period requested and then grouped by application, invoking user, host name, transaction name and date. A new graph and detail report is shown for each unique combination of transaction, user, agent and application.

The average response time is calculated as shown in the following calculation:

$$\text{sum}(\text{avg_response_time} * \text{transaction_execution_count}) / \text{sum}(\text{transaction_execution_count})$$

This translates to the following fact table columns:

$$\text{sum}(\text{fact_table.sample_count} * \text{fact_table.sample_count}) / \text{sum}(\text{fact_table.sample_count})$$

The minimum response time is calculated as shown in the following calculation:

$$\text{min}(\text{avg_response_time})$$

This translates to the following fact table columns:

`min(fact_table.min_value)`

3.2.16.1 Start Date and End Date

The Start_Date and End_Date input parameter values are chosen from a pop-up calendar before running the report so all data retrieved is greater than or equal to the Start_Date value and less than or equal to the End_Date value. Only one value is permitted for each parameter. When the WEEK time granularity is chosen the data is dated for the beginning Sunday of each week and this date must fall on or between the Start and End dates to be shown. When the MONTH time granularity is chosen the data is dated for the first day of each month and this date must fall on or between the Start and End dates to be shown.

3.2.16.2 Time Granularity

The time granularity input parameter is a drop-down menu where you can select from one of HOUR, DAY, WEEK, or MONTH. This parameter is used to select from which fact table the data will be read.

3.2.16.3 Agent Name

The agent name, or hostname, input parameter can be a single, specific policy name or can be wild carded. To get all policy names use the '*' wild card value or, to get a set of policy names, use wild carding like AC* or *AC*. If there are a large number of agents to show on the report the graph may not be able to scale to show all the agent names, so only a reasonable quantity should be selected for viewing.

3.2.16.4 Application Name

The application name input parameter can be a single application name or can be wild carded. To get all applications use the '*' wild card value or to get a set of applications use wild carding like AC* or *AC*. This input parameter is a text field and only characters should be entered for the application name.

3.2.16.5 User Name

The user name input parameter can be a single, specific user name or can be wild carded. To get all user names use the '*' wild card value or, to get a set of user names, use wild carding like AC* or *AC*. If there are a large number of users to show on the report the graph may not be able to scale to show all the user names, so only a reasonable quantity should be selected for viewing.

3.2.16.6 Transaction Name

The transaction name input parameter can be a single, specific transaction name or can be wild carded. To get all transaction names use the '*' wild card value or, to get a set of transaction names, use wild carding like AC* or *AC*. If there are a large number of transactions to show on the report the graph may not be able to scale to show all the transaction names, so only a reasonable quantity should be selected for viewing.

3.2.16.7 Number of Transactions

The 'Maximum Number of Transactions' input parameter defines the limit of how many transactions to show in the report. The transactions with the lowest execution count will be shown in ascending order.

3.2.17 Enhanced Availability

For this report various transaction count trends are shown over the time period requested and then grouped by application. The transaction counts are calculated as shown in the following calculation:

Total transaction count = `sum(transaction_count)`

Transaction_good_count = sum(transaction_good_count - transaction_error_count - transaction_abort_count)

Transaction_abort_count = sum(transaction_abort_count)

Transaction_error_count = sum(transaction_error_count)

Transaction_violation_count = sum(transaction_violation_count)

This translates to the following fact table columns:

Total transaction count = sum(fact_table.sample_count (where metric_id = 'number of executions') + fact_table.sample_count (where metric_id = 'number of aborted transactions'))

Total_good_count = sum(fact_table.sample_count - fact_table.error_count)
(where metric_id = 'number of executions'))

Total_abort_count = sum(fact_table.error_count)
(where metric_id = 'number of aborted transactions'))

Total_error_count = sum(fact_table.error_count)

Total_violation_count = sum(fact_table.sample_count)
(where metric_id = 'number threshold exceeded'))

3.2.17.1 Start Date and End Date

The Start_Date and End_Date input parameter values are chosen from a pop-up calendar before running the report so all data retrieved is greater than or equal to the Start_Date value and less than or equal to the End_Date value. Only one value is permitted for each parameter. When the WEEK time granularity is chosen the data is dated for the beginning Sunday of each week and this date must fall on or between the Start and End dates to be shown. When the MONTH time granularity is chosen the data is dated for the first day of each month and this date must fall on or between the Start and End dates to be shown.

3.2.17.2 Time Granularity

The time granularity input parameter is a drop-down menu where you can select from one of HOUR, DAY, WEEK, or MONTH. This parameter is used to select from which fact table the data will be read.

3.2.17.3 Application Name

The application name input parameter can be a single application name or can be wild carded. To get all applications use the '*' wild card value or to get a set of applications use wild carding like AC* or *AC*. This input parameter is a text field and only characters should be entered for the application name.

3.2.18 Resource Usage, Response Time and Execution Load For Transaction <transaction>

For this report to run the following requirements must be met:

- The IBM Tivoli Monitoring AMX and AMY warehouse packs must be installed and contain data that contains some of the same host names and measurement dates as the Tivoli Monitoring for Transaction Performance data.
- The IBM Tivoli Monitoring AMY warehouse pack must exist in the same data mart database as the Tivoli Monitoring for Transaction Performance BWM warehouse pack.

- The script, `Create_Cross_App_report_views.sql`, found in the `$TWH_TOPDIR/apps/bwm/v5300/misc/tools` directory must have been run to create the required database views the reports use to collect data.

For this report's first graph, the transaction average response time trend is shown over the time period requested for the transaction specified and then grouped by host name and date. This is contrasted, on the same graph, with the CPU and memory trends for the same time period and hosts.

The average response time is calculated as shown in the following calculation:

$$\text{sum}(\text{avg_response_time} * \text{transaction_execution_count}) / \text{sum}(\text{transaction_execution_count})$$

This translates to the following fact table columns:

$$\text{sum}(\text{fact_table.sample_count} * \text{fact_table.sample_count}) / \text{sum}(\text{fact_table.sample_count})$$

For this report's second graph, the transaction execution count trend is shown over the time period requested for the transaction specified and then grouped by host name and date. This is contrasted, on the same graph, with the CPU and memory trends for the same time period and hosts.

The transaction execution count is calculated as shown in the following calculation:

$$\text{sum}(\text{transaction_execution_count})$$

This translates to the following fact table columns:

$$\text{sum}(\text{fact_table.sample_count})$$

A new graph and detail report is shown for each unique combination of transaction and host name.

3.2.18.1 Start Date and End Date

The `Start_Date` and `End_Date` input parameter values are chosen from a pop-up calendar before running the report so all data retrieved is greater than or equal to the `Start_Date` value and less than or equal to the `End_Date` value. Only one value is permitted for each parameter. When the WEEK time granularity is chosen the data is dated for the beginning Sunday of each week and this date must fall on or between the Start and End dates to be shown. When the MONTH time granularity is chosen the data is dated for the first day of each month and this date must fall on or between the Start and End dates to be shown.

3.2.18.2 Time Granularity

The time granularity input parameter is a drop-down menu where you can select from one of HOUR, DAY, WEEK, or MONTH. This parameter is used to select from which fact table the data will be read. For this report, the value chosen applies to both the Tivoli Monitoring for Transaction Performance fact tables and the AMY fact tables so the same time granularity is consistent across fact tables for data merging.

3.2.18.3 Agent Name

The agent name, or hostname, input parameter can be a single, specific policy name or can be wild carded. To get all policy names use the '*' wild card value or, to get a set of policy names, use wild carding like `AC*` or `*AC*`. If there are a large number of agents to show on the report the graph may not be able to scale to show all the agent names, so only a reasonable quantity should be selected for viewing.

3.2.18.4 Transaction Name

The transaction name input parameter can be a single, specific transaction name or can be wild carded. To get all transaction names use the '*' wild card value or, to get a set of transaction names, use wild carding like AC* or *AC*. If there are a large number of transactions to show on the report the graph may not be able to scale to show all the transaction names, so only a reasonable quantity should be selected for viewing.

3.3 Report Notes and Limitations

3.3.1 Viewing reports

- When selecting how much data to show in a report, consider the space available for the graph in the web browser and limit the Start Date and End Date fields appropriately to optimize graphics. Only show up 30 data points at a time; excessive amounts of data to plot in one graph may render the graph difficult to read or cause some of the chart objects to overlap and become unreadable.
- In some cases, the report's graph does not fit on the first page of the report. When this occurs, the first page of the report only shows the report criteria box, then the report's graph and data show on the second and subsequent pages. The first page number seen for the report is on the page with the graph or first page of the report and is numbered with a '2'. Look for the '1+' indicator at the top of the first report page shown to see that there are additional report pages. For more information on this functionality visit web page:
<http://support.businessobjects.com/library/kbase/articles/c2014404.asp>
- Depending on the time granularity chosen for the report, the X-axis date column will show the date and time values as follows:
 - HOUR - shows the date and hour of the data
 - DAY - shows the day of the data
 - WEEK - shows the date of each Sunday for the weeks shown
 - MONTH - shows the month and year
- In order for the time zone information to show correctly in the reports, set the time zone in Crystal's ePortfolio preferences. If this is not set, the 'System Default Timezone' will be shown instead.
- There is no drill-down capability in this warehouse pack's reports. Instead, use the navigation bar on the left-hand side of most graphs to navigate to the detail data.
- Reports that process large amounts of data may run for long periods of time, may time out or may show the error:

"There was an error retrieving data from the server: Max processing time/Max records limit reached".

In these cases it may be preferable to *schedule* the reports instead of *viewing* the reports because the Crystal page server process may time out when the *view* option waits for the database sql to return all the data. By default, 20,000 records is the limit before this error message is shown. If you'd like to change the limit, please read the article on this web site for how to change the default value.

<http://support.businessobjects.com/library/kbase/articles/c2015524.asp>

3.3.2 Report fonts

In order to support languages other than English, you must install the Monotype Sans Duospace WT font on the system that is the Crystal Enterprise server. If you will be supporting DBCS languages, you must also install the following additional variants of this font on the Crystal Enterprise server:

- Monotype Sans Duospace WT J - for Japanese
- Monotype Sans Duospace WT K - for Korean
- Monotype Sans Duospace WT SC - for Simplified Chinese
- Monotype Sans Duospace WT TC - for Traditional Chinese

Obtain the Monotype Sans Duospace WT font, and any necessary font variants needed for DBCS languages. The *Monotype Sans Duospace WT J* font is included on our product CD in the `tdw_fonts` directory. Copy the font or fonts to the `x:\WINNT\Fonts` directory of the system that is the Crystal Enterprise server, where `x` is the system drive. You will need to reboot your system in order for the fonts to be used.

In most cases, the fonts will be replaced by the default system font on systems where reports are being viewed. However, these fonts must be installed on the Crystal Enterprise server to enable support of DBCS languages that are embedded in reports, such as the PDF format. For example, if a Japanese string is embedded in a report as a bitmap, the string will not display correctly on the system where you are viewing the report unless these fonts are installed on the Crystal Enterprise server where the bitmap is stored.

3.3.3 Limitations

- The reports using time granularity of DAY, WEEK, or MONTH and calculate or show data using the `error_count` column in the fact tables may have inaccurate or missing data due to the warehouse rollup step not currently rolling up the `error_count` column. Refer to section 4.2.2.4 for more details on this problem.
- When viewing the report parameter page in the German language, the 'Select' buttons are misspelled. This is a known issue, addressed in Crystal incident number 1987629, that will be fixed in a future release.
- In some languages a few report column headers may wrap to a new line in the middle of a word. This is a current limitation that will be addressed in future Crystal release.

4 Installing and configuring the warehouse pack

This section describes the installation and configuration of the warehouse pack.

4.1 Prerequisite hardware and software

Before installing the warehouse pack for IBM Tivoli Monitoring for Transaction Performance, Version 5.3.0, you must install the following software:

- IBM Tivoli Monitoring for Transaction Performance Version 5.3
 - IBM DB2 Universal Database, Version 7.2
 - See the *Tivoli Data Warehouse, Version 1.2 Release Notes* for the fix pack level of DB2 required.
 - Tivoli Data Warehouse, Version 1.2
 - Tivoli Data Warehouse, Version 1.2 Fix Pack 2
 - Crystal Enterprise and its prerequisites
 - Crystal Enterprise patch for Crystal Enterprise 9 Track ID ADAPT00319691 (See section 4.1.1 below for more details)
 - (Optional) IBM Tivoli Service Level Advisor™ 1.2.1 Fix Pack 1
- Note: Read the documentation for the IBM Tivoli Service Level Advisor™ 1.2.1 warehouse pack to be familiar with its requirements before installing this warehouse pack.
- IBM Tivoli Monitoring for Transaction Performance warehouse pack 5.2.0, if you are upgrading to this version of the warehouse pack
 - When installing the warehouse pack in a distributed machine environment, it may be necessary to increase the DB2 APPLHEAPSZ parameter for the warehouse pack to install successfully.

This warehouse pack supports central data warehouses and data marts on DB2 UDB for Windows and UNIX systems, but not DB2 UDB for z/OS and OS/390.

Refer to the *Tivoli Data Warehouse Release Notes* and *IBM Tivoli Monitoring for Transaction Performance, Version 5.3 Release Notes* for specific information about hardware prerequisites, database and operating system support, and product prerequisites. For late-breaking news about prerequisites, refer to the following IBM Software Support Web site:

<http://www.ibm.com/software/sysmgmt/products/support/>

4.1.1 Installing the Crystal Enterprise 9 patch for ADAPT00319691

This patch fixes a problem where reports written using SQL command objects cannot open properly and fail before the parameter input page is opened. Follow the directions found at the web site below to install this patch before using the reports included with this warehouse pack.

For a more detailed synopsis of the defect and the instructions to apply the patch, visit web page <http://support.businessobjects.com/library/kbase/articles/c2016181.asp>.

4.2 Product notes and limitations

4.2.1 Product notes

The warehouse pack for IBM Tivoli Monitoring for Transaction Performance 5.3 supports all versions of DB2 and Oracle database products as documented in the *IBM Tivoli Monitoring for Transaction Performance Release Notes*, GC23-4803-00.

4.2.2 Limitations

4.2.2.1 No support for DB2 Universal Database for z/OS and OS/390

This warehouse pack does not support running this warehouse pack on DB2 Universal Database for z/OS and OS/390. Read the documentation for the IBM Tivoli Service Level Advisor™ 1.2.1 warehouse pack to be familiar with its requirements before installing this warehouse pack.

4.2.2.2 Synthetic Edge Transactions in the warehouse

Root transactions that are defined in the Tivoli Monitoring for Transaction Performance console as patterns that should be matched to particular WebSphere or WebLogic servers are not stored in the central data warehouse database with the same relations as the other J2EE_SERVER components. Examples of the root transaction patterns are J2EE/.*, J2EE/WebSphere/4.0.1/.*. These edge transactions are stored as BWM_TX_NODE components and have measurements associated with them but they are not stored as J2EE_SERVER components. In addition these BWM_TX_NODE components will not have the traditional BWM_PROBE or J2EE_SERVER relations to show the application related to the node.

4.2.2.3 Changing the ETL source input database

If the Tivoli Monitoring for Transaction Performance source database that the ETL extracts data from is changed to point to a different Tivoli Monitoring for Transaction Performance source database the `Clean_and_Reset_BWM_warehouse_Delete_tables.sql` and `Clean_and_Reset_BWM_warehouse_Create_tables.sql` scripts will need to be run prior to uninstalling and reinstalling this warehouse pack. See sections 5.4.11 and 5.4.12 for more details on these scripts.

If a different source input database is used or the current Tivoli Monitoring for Transaction Performance source database is reinitialized, the sequence number keys of the database tables will cause conflicts with the data already stored in the central data warehouse database that already refer to sequence numbers causing data integrity issues in the warehouse. To switch to a new Tivoli Monitoring for Transaction Performance source database it is required that the `Clean_and_Reset_BWM_warehouse_Delete_tables.sql` and `Clean_and_Reset_BWM_warehouse_Create_tables.sql` scripts be run to remove the current data for this warehouse pack from the central data warehouse database. Only SHARED components and their non-BWM relations will be left in the central data warehouse database.

If it is desired to remove the SHARED data from the central data warehouse database the remaining data will need to be manually removed.

4.2.2.4 Rollup of Error_count data from the hourly fact table in the MART database

The `error_count` column of the `BWM.F_TX_ND_HOUR` table is currently not rolled up into the daily, weekly, and monthly fact tables. In these tables it has a NULL value. The fix for this will be available in the near future.

4.2.2.5 Reinstalling the Tivoli Monitoring for Transaction Performance warehouse pack

When you uninstall an IBM Tivoli Monitoring for Transaction Performance warehouse pack, IBM Tivoli Monitoring for Transaction Performance data in the central data warehouse is not automatically removed. This is because data in the central data warehouse is not tied to a specific application. It is a historical record of activity and conditions in your enterprise. Although it might have been placed in the Tivoli Data Warehouse by one application, it can be read by other applications, including those you write yourself. Therefore, Tivoli Data Warehouse does not provide an automated way to delete data from the central data warehouse, aside from the data pruning mechanisms.

If you do not delete the data, a subsequent reinstallation of an IBM Tivoli Monitoring for Transaction Performance warehouse pack can lead to a potential duplication of data in the central data warehouse. Attempting to insert duplicates into the Tivoli Data Warehouse results in SQL errors and the ETL process ends processing on errors. To avoid this situation, do the following after you uninstall an ETL:

1. Set `PMsmtC_Age_In_Days = -1` in the `TWG.PRUNE_MSMT_CONTROL` table. See section 5.2.1.1 for details.
2. Set `PMartC_Duration = -1` in the `BWM.Prune_Mart_Control` table. See section 5.2.2.1 for details. The `PMsmtC_Age_In_Days` and `PMartC_Duration` values are set to -1 because the purge process does not delete the data for the current day and this value allows it to remove data from the current day.
3. Run the Tivoli Data Warehouse `CDW_c05_Prune_and_Mark_Active` process.
4. Reinstall the IBM Tivoli Monitoring for Transaction Performance warehouse pack.

4.2.2.6 How to handle multiple failed unattended executions of the ETL

If the central data warehouse ETL runs on Day 1, fails on the last step, but then continues to be re-run on successive days without the problem causing the failure having been resolved, there may be a loss of extracted data for the time period that the failure continued occurring. After fixing the source of the original failure, the Tivoli Data Warehouse user needs to update the extract control information to pick up data from the point of failure and rerun the ETL process. Be aware if you gather large amounts of data daily or the problem was left unattended for a long time that re-extracting the data missed during the ETL failures may take a significant amount of time and database resources. Also, note that if the Tivoli Monitoring for Transaction Performance source database had data purged after the ETL failure occurred but before the failure was corrected and the ETL rerun, then the re-extraction of data will not be able to include the data that was removed. It is recommended that the ETL processes be monitored for success or failure after each run so the problem can be corrected in a timely manner.

If the problem occurs follow the steps below to re-extract all the data since the beginning of the failures, assuming the data has not been purged from the source database.

Step 1: Get a list of the times and sequences collected for each run of the warehouse pack ETL. Run the `misc/tools/Show_ExtLog_Values.sql` script from the DB2 Universal Database command line processor.

```
db2 -tvf Show_ExtLog_Values.sql
```

Or run the following DB2 Universal Database command from the command line processor after connecting to the central data warehouse as a valid, authorized DB2 Universal Database warehouse user.

```
SELECT ExtLog_Source, ExtLog_DONE_DTTM, ExtLog_FROM_INTSEQ, ExtLog_TO_INTSEQ
FROM TWG.Extract_Log
WHERE ExtLog_TARGET='BWM.STAGE_AGGREG_DATA'
```

Sample output of above command:

EXTLOG_SOURCE	EXTLOG_DONE_DTTM	EXTLOG_FROM_INTSEQ	EXTLOG_TO_INTSEQ
1.AGGREGATEDATA	2004-02-26-01.30.09.250000	-1	109950
1.AGGREGATEDATA	2004-02-26-01.45.03.562001	109950	109950
1.AGGREGATEDATA	2004-02-27-01.30.14.687000	109950	119696
1.AGGREGATEDATA	2004-02-27-01.45.04.562001	119696	119696
1.AGGREGATEDATA	2004-02-28-01.30.14.515000	119696	130068
2.AGGREGATEDATA	2004-02-28-01.45.04.953000	130068	130068
2.AGGREGATEDATA	2004-02-29-01.30.15.531000	130068	137044
2.AGGREGATEDATA	2004-02-29-01.45.05.796001	137044	137044
2.AGGREGATEDATA	2004-03-01-01.30.15.250000	137044	142452

Step 2: Find the date for which the initial failure occurred and view the Extract Log's value in column ExtLog_FROM_INTSEQ.

Step 3: Update the Extract_Control table's from integer sequence column, ExtCtl_FROM_INTSEQ, with the value from the Extract Log's column, ExtLog_FROM_INTSEQ, you selected in step 2. Use the following SQL statement to update the Extract Control table.

```
UPDATE TWG.Extract_Control
SET ExtCtl_FROM_INTSEQ = sequence_number
WHERE ExtCtl_Target = 'BWM.STAGE_AGGREG_DATA'
AND ExtCtl_Source = '1.AGGREGATEDATA';
```

Where *sequence_number* is the sequence number you selected. It is not necessary to update the Extract Control table's column, ExtCtl_TO_INTSEQ, as the central data warehouse process automatically sets this value to the current highest sequence number found in the AggregateData source table.

4.3 Database-sizing considerations

Ensure that you have sufficient space in the central data warehouse for the historical data collected by the warehouse pack. Refer to the following worksheet as an example of database sizing considerations for IBM Tivoli Monitoring for Transaction Performance Version 5.3.0 warehouse pack.

Database	Schema	Tables	Table Row Size (byte)	Table Size per 1K rows (Mb, est.)	Table Size per 1K rows (Mb, min)	Table Size per 1K rows (Mb, max)	Index Size per 1K rows (Mb, est.)
TWH_MART	BWM	D_TX_ND_METRIC	668	0.86	0.04	1.43	0.02
TWH_MART	BWM	D_HOST	531	0.61	0.03	1.43	0.02
TWH_MART	BWM	D_APP	796	0.86	0.04	2.15	0.02
TWH_MART	BWM	D_TX	1558	2.15	0.07	4.30	0.02
TWH_MART	BWM	D_TX_ND	532	0.61	0.03	1.43	0.02
TWH_MART	BWM	F_TX_ND_HOUR	92	0.11	0.11	0.11	0.23
TWH_MART	BWM	F_TX_ND_DAY	92	0.11	0.11	0.11	0.23
TWH_MART	BWM	F_TX_ND_WEEK	92	0.11	0.11	0.11	0.23
TWH_MART	BWM	F_TX_ND_MONTH	92	0.11	0.11	0.11	0.23
TWH_MART	BWM	PRUNE_MART_CONTROL	69	0.08	0.02	0.14	-
TWH_MART	BWM	PRUNE_MART_LOG	40	0.05	0.03	0.06	-
TWH_MART	BWM	STG_F_TX_ND_HR	100	0.11	0.11	0.11	0.09
TWH_CDW	TWG	COMP	453	0.54	0.09	1.07	0.02
TWH_CDW	TWG	COMPATTR	176	0.20	0.06	0.36	0.09

TWH_CDW	TWG	COMPRELN	38	0.05	0.05	0.05	0.08
TWH_CDW	TWG	MSMT	70	0.08	0.08	0.08	0.08

4.4 Pre-installation procedures

The IBM Tivoli Monitoring for Transaction Performance warehouse pack 5.2.0 must be installed before upgrading to this version, version 5.3.0.

4.4.1 Tivoli Data Warehouse 1.2 required

Install or upgrade the Tivoli Data Warehouse to version 1.2 following the instructions in the *Installing and Configuring Tivoli Data Warehouse, Version 1.2* guide before installing or upgrading to this version of the Tivoli Monitoring for Transaction Performance warehouse pack. This warehouse pack will only run on Tivoli Data Warehouse, Version 1.2. When you upgrade to Tivoli Data Warehouse, Version 1.2 any currently installed warehouse pack will be automatically upgraded to the Version 1.2 format as well but with only Tivoli Data Warehouse, Version 1.1 functionality. When a Version 1.2 warehouse pack is installed after upgrading from Tivoli Enterprise Data Warehouse, Version 1.1 to Tivoli Data Warehouse, Version 1.2, the changes required to make the warehouse pack fully functional are installed.

4.4.2 Make warehouse database backups

Before installing or upgrading to a new warehouse pack, backups should be made of the warehouse databases TWH_CDW, TWH_MD and TWH_MART as well as the \$TWH_TOPDIR/apps directory to establish a point of recovery in case of installation failures.

4.4.3 Upgrading to the new Web Transaction Performance 5.3.0.0 warehouse pack

When upgrading to this new warehouse pack, the data mart star schema will be modified to include the new agent group, management policy and management policy group data in the star schema. To accomplish this, the upgrade installation process renames the current dimension tables and creates new ones, with the new columns, to replace them. Once the new dimension tables are created, the data from the renamed tables is copied one table at a time into the new dimension tables. This prevents a loss of data. Because the dimension tables can contain large amounts of data, the transaction logs must be sized correctly to enable the data to be inserted from the old dimension tables into the new ones without running out of transaction logs. Consider increasing the size and number of the Tivoli Data Warehouse data mart transaction logs. If the transaction logs are sized to work for the largest of the dimension tables, all will work because automatic commits occur after each SQL statement is completed.

Before upgrading to this warehouse pack, remove the Crystal reports from the Crystal Enterprise server that were installed with the prior version of this warehouse pack. The reports from the previous version do not run with this warehouse pack.

4.4.4 ODBC database client

If the Tivoli Monitoring for Transaction Performance source database and Tivoli Data Warehouse do not use the same database vendor type, you must install a database client for the Tivoli Monitoring for Transaction Performance on the Tivoli Data Warehouse machine. For example, if the Tivoli Monitoring for Transaction Performance database is implemented on an Oracle database server, then the Oracle client must be installed on the Warehouse machine. This is so the ODBC connection can use the client to communicate with the Tivoli Monitoring for Transaction Performance source database.

4.4.5 ODBC drivers

See the getting started information in the *Installing and Configuring Tivoli Data Warehouse, Version 1.2* guide to create a System DSN entry using the ODBC driver that is appropriate to the Tivoli Monitoring for Transaction Performance database vendor:

DataWHSE 3.60 32-bit Oracle8

IBM DB2 ODBC DRIVER

These drivers are installed with the installation of any version later than and including DB2 Universal Database, Version 7.2. The driver used with DB2 Universal Database, Version 7 can also be used with Version 8. The driver used with Oracle 8 must also be used with Oracle 9. Do not use the Oracle 9 ODBC driver as this is not supported by Tivoli Data Warehouse, Version 1.2 and will not connect properly through the Tivoli Data Warehouse.

TMTP_DB_SRC is the default Tivoli Monitoring for Transaction Performance ODBC data source name in the BWM_TMTP_DB_SRC_Source warehouse source setting.

4.5 Installation of the warehouse pack

Before installing this warehouse pack, record the user IDs, passwords, and database server name used to connect to the Tivoli Monitoring for Transaction Performance database in the following table. You need this information to follow the installation procedures that are described in *Installing and Configuring Tivoli Data Warehouse*. If you have multiple Tivoli Monitoring for Transaction Performance data sources, complete this information for each data source.

ODBC source	User ID	Password	Database type	Server name
The default data source name for the ODBC connection is TMTP_DB_SRC.	This is the user id to access the Tivoli Monitoring for Transaction Performance database using the ODBC connection	This is the password used to access the ETP database for the user specified in the previous column.	DB2 UDB, Oracle	

Install the warehouse pack as described in *Installing and Configuring Tivoli Data Warehouse*, using the installation properties file (tw_h_install_props.cfg file). This file is located in tw_weps directory in the IBM Tivoli Monitoring for Transaction Performance, Version 5.3 CD

4.6 Post-installation procedures

4.6.1 One time process to run to upgrade warehouse data

See section 6.1, BWM_c05_Upgrade_Process, for details on the one time upgrade process to be run when upgrading to this warehouse pack.

4.6.2 Scheduling warehouse pack processes

The warehouse pack extract schedule may be set during the installation of the warehouse pack but the extract processing will not run until the one-time process, BWM_c05_Upgrade_Process, has been run successfully. You must run this upgrade process, if you have upgraded to this version of the warehouse pack to upgrade the warehouse central data warehouse and DATA MART data and tables. After each step of this process has run to successful completion you should then promote the three central data warehouse ETL (ETL1) steps and five data mart ETL (ETL2) steps to 'production' mode. After this is done the extraction process will run on the schedule selected during the warehouse pack installation. If

not schedule was selected during the warehouse pack installation process this will need to be done before the extraction process will run.

The data mart ETL (ETL2) extract runs automatically after central data warehouse ETL (ETL1) successfully completes. If, after you installed the warehouse pack, you want to reschedule the warehouse pack extract schedule see the information about installing warehouse packs in *Installing and Configuring Tivoli Data Warehouse* for the procedure to schedule ETLs. Use these process dependencies when changing any extract scheduling:

Initialization process	none
Process dependencies	Located in the BWM_Tivoli_Monitoring_for_Transaction_Performance_v5.3.0.0_Subject_Area subject area
	The processes should be run in the following order:
	1. BWM_c05_Upgrade_Process
	2. BWM_c10_src_to_cdw_Process
	3. BWM_m05_cdw_to_mart_Process

4.6.3 Change Prune Control values from installed defaults

The warehouse pack installs with the prune control values already configured with default values. Measurement data is pruned after it has been in the Tivoli Data Warehouse more than 3 months. Hourly and daily data mart data is pruned after it has been in the data mart more than 3 months and weekly and monthly data is pruned after it has been in the data mart more than 1 year. See section 5.2, Deleting Data, for more details on changing these values.

The first time the ETL process is run it retrieves only the data that is greater than the prune measurement value set for central data warehouse so that data is not needlessly extracted on the first ETL run that will be pruned at the end of the run. Not extracting the undesired data will improve the performance of the initial extraction and pruning processes.

4.6.4 Accessing the application source database

The source database for the ETL is the database in which IBM Tivoli Monitoring for Transaction Performance is storing data. IBM Tivoli Monitoring for Transaction Performance, Version 5.3 supports database servers for Oracle and DB2. This warehouse pack supports the same database server versions as IBM Tivoli Monitoring for Transaction Performance.

Before you define the Tivoli Data Warehouse source, you must verify that the IBM Tivoli Monitoring for Transaction Performance database exists and that you can connect to it. For DB2 Universal Database, you must catalog the database with ODBC. For Oracle, you must supply the system data source name (DSN) by which the database is registered in ODBC on the agent site. Use the "Data Sources (ODBC)" program, accessible from the Windows NT Control Panel, to add the source data source as a system data source. The ODBC driver to be used for DB2 is "IBM DB2 ODBC DRIVER" Version 7.01.00.88 from IBM. The ODBC driver to be used for Oracle is "DataWHSE 3.60 32-bit Oracle8" Version 3.60 from MERANT. This driver should be used with Oracle 8 and Oracle 9. Do not use the Oracle 9 ODBC driver as this is not supported by Tivoli Data Warehouse 1.2 and will not connect properly through the Tivoli Data Warehouse.

TMTTP_DB_SRC is the default Tivoli Monitoring for Transaction Performance ODBC data source name in the BWM_TMTTP_DB_SRC_Source warehouse source setting.

The Oracle TIME_ZONE must be in numeric format, not the name equivalent – i.e, '-07:00' and not 'CST'. To see and change this, run these commands:

```
SELECT DBTIMEZONE FROM DUAL;
```

```
ALTER DATABASE SET TIME_ZONE = '-07:00';
```

4.6.5 Configure warehouse source and target settings

The following sources and targets are created by the IBM Tivoli Monitoring for Transaction Performance, Version 5.3.0 warehouse pack:

- BWM_TWH_CDW_Source: a source object for the central data warehouse
- BWM_TWH_MART_Source: a source object for the TWH_MART database
- BWM_TMTP_DB_SRC_Source: a source object for the IBM Tivoli Monitoring for Transaction Performance source database
- BWM_TWH_CDW_Target: a target for the central data warehouse
- BWM_TWH_MART_Target: a target for the TWH_MART database

The warehouse pack is installed by Tivoli Data Warehouse 1.2 with the warehouse sources and target values already set. If, after installing the warehouse pack, you need to change any of the source or target properties because IDs or passwords have changed use the procedures in *Installing and Configuring Tivoli Data Warehouse* to perform the following configuration tasks for data sources and targets:

1. Specify the properties for the BWM_TMTP_DB_SRC_Source data source, ODBC Source.
 - Set Data source name (DSN) to the name of the ODBC connection for the BWM_TMTP_DB_SRC_Source. The default value is TMTP_DB_SRC.
 - Set the User ID field to the user ID used to access the BWM_TMTP_DB_SRC_Source. The default value is *db2admin*.
 - Set the Password field to the password used to access the BWM_TMTP_DB_SRC_Source.
2. Specify the properties for the target BWM_TWH_CDW_Source.
 - In the User ID field, type the user ID used to access the Tivoli Data Warehouse central data warehouse. The default value is *db2admin*.
 - In the Password field, type the password used to access the central data warehouse database.

Do not change the value of the Database Name field. It must be TWH_CDW.

3. Specify the following properties for the target BWM_TWH_MART_Source.
 - In the User ID field, type the user ID used to access the data mart database. The default value is *db2admin*.
 - In the Password field, type the password used to access the data mart database.
 - Do not change the value of the Database Name field. It must be TWH_MART.
4. Specify the properties for the warehouse target BWM_TWH_CDW_Target.
 - In the User ID field, type the user ID used to access the central data warehouse database. The default value is *db2admin*.
 - In the Password field, type the password used to access the central data warehouse database.
 - Do not change the value of the Database Name field. It must be TWH_CDW.
5. Specify the following properties for the target BWM_TWH_MART_Target.
 - In the User ID field, type the user ID used to access the data mart database. The default value is *db2admin*.
 - In the Password field, type the password used to access the data mart database.

- Do not change the value of the Database Name field. It must be TWH_MART.

4.7 Migration from a previous release of the warehouse pack

The following database objects are changed since the previous release of the warehouse pack.

Added objects:

- Additional reports
- BWM_AGENT_GRP CompTyp
- BWM_MGMT_POLICY CompTyp
- BWM_MGMT_POL_GRP CompTyp
- Exception logging tables

Changed objects:

- Prior reports have been redone
- Staging tables have been renamed to follow the pattern STG_* instead of STAGE_*
- ETL process and step names have been renamed to conform to Tivoli Data Warehouse 1.2 naming standards

Removed objects:

- None

4.8 Uninstalling the warehouse pack

Perform the following steps to uninstall the warehouse pack:

1. Uninstall the warehouse pack as described in the *Installing and Configuring Tivoli Data Warehouse* guide. **Remember to manually remove the Crystal reports.**

When the warehouse pack is uninstalled, the following staging tables are removed, but the data in the central data warehouse remains and is still useable by other applications:

Views

BWM.VE_HOST
 BWM.VD_HOST
 BWM.VE_APP
 BWM.VE_TX
 BWM.VD_TX
 BWM.VE_TX_ND_MET
 BWM.VD_TX_ND_MET
 BWM.VE_TX_ND
 BWM.VE_STG_TX_ND_HR
 BWM.VE_COMP_NAME_LONG
 BWM.VD_COMP_NAME_LONG
 BWM.COMP_NAME_LONG
 BWM.VE_COMP_ATTR_LONG

BWM.VD_COMP_ATTR_LONG

Staging Tables

BWM.STG_TX_ND_MET

BWM.COMP_NAME_LONG

BWM.COMP_ATTR_LONG

BWM.CENTR_LOOKUP

BWM.CUST_LOOKUP

BWM.STAGE_HOST

BWM.STAGE_AGENT_GRP

BWM.STAGE_APPLICATION

BWM.STAGE_USER

BWM.STAGE_TRANSACTION

BWM.STAGE_NODE

BWM.STAGE_RELATIONMAP

BWM.STAGE_AGGREG_DATA

BWM.STAGE_MGMTPOLICY

BWM.STAGE_MGMTPOLGRP

BWM.STAGE_PATTERN

BWM.STAGE_THRESHOLD

BWM.STAGE_TR

Sequences

BWM.THR_MSMT_ID_SEQ

BWM.FACT_ID_SEQ_HR

BWM.FACT_ID_SEQ_DY

BWM.FACT_ID_SEQ_WK

BWM.FACT_ID_SEQ_MN

4.9 Multiple data centers

After you install the warehouse pack, you can configure Tivoli Data Warehouse to separate data for multiple data centers. To set this up, you must create SQL scripts with the following values:

Information for scripts	Value or location
Field in source data	Fully qualified host name
Name of lookup table	BWM.Centr_lookup table
Name of center list	TWG.Centr

For the procedural instructions and sample SQL statements, see the information in *Enabling an Application for Tivoli Data Warehouse*.

After the initial configuration for multiple data centers, you must modify the tables when data centers are added and removed

4.10 Multiple customer environments

After you install the warehouse pack, you can configure Tivoli Data Warehouse to separate data for the multiple customer environments. To set this up, you must create SQL scripts with the following values:

Information for scripts	Value or location
Field in source data	Fully qualified host name
Name of lookup table	BWM.Cust_lookup table
Column to use for lookup	Cust_ID
Name of customer list	TWG.Cust

For the procedural instructions and sample SQL statements, see the information in *Enabling an Application for Tivoli Data Warehouse*.

After your initial configuration of the multiple customer environments, you must modify the tables when customers are added and removed.

5 Maintenance and problem determination

This section describes maintenance tasks for the warehouse pack.

5.1 Backing up and restoring

This section describes additional information about backing up and restoring data for the warehouse pack.

To back up a database from the DB2 Command Line Processor do:

```
mkdir <backup_dir_name>
cd <backup_dir_name>
db2stop force
db2start
db2 backup db <database_name>
```

The DB2 GUI can be used to back up databases as well.

See *Installing and Configuring Tivoli Data Warehouse, Version 1.2* for more information about backing up and restoring Tivoli Data Warehouse databases.

5.2 Pruning data

To manage the high volume of warehouse data, use pruning processes to remove data no longer required.

5.2.1 Central data warehouse

To manage the high volume of measurement data, use the TWG.Prune_Msmt_Control table settings to remove older data. By default, data in the Msmt table older than 3 months is deleted when the CDW_c05_Prune_and_Mark_Active process runs. This process is within the CDW_Tivoli_Data_Warehouse_v1.2.0_Subject_Area.

By default, this process runs daily at 6:00 a.m. If this schedule is not appropriate for you, schedule the time and interval - daily, weekly, or monthly - you want this process to run.

The TWG.Prune_Msmt_Log table keeps a history of the range of measurement data removed.

5.2.1.1 Pruning measurement data (table Prune_Msmt_Control)

Measurement data is deleted from the Msmt table every 3 months. This is based on the age specified in the PMSmtC_Age_In_Days column of the Prune_Msmt_Control table for this warehouse pack. You can modify this value by running the following SQL statement, where *X* is a date duration whose format is *yyyymmdd* (for example: *X* = 00000108 for 0000 years, 01 month, 08 days).

Connect to TWH_CDW

```
UPDATE TWG.Prune_Msmt_Control
  SET PMSMTC_AGE_IN_DAYS = X
  WHERE TMSUM_CD = 'H' AND MSRC_CD = 'BWM'
```

See section 5.4, Other Maintenance Tools, for information on scripts that will set prune values for you.

5.2.2 Data mart

Deleting data from the data mart fact tables is implemented in the BWM_m05_s050_mart_prune process. The prune data mart control table governs which data is deleted based on the duration value set in the PMartC_Duration column. This warehouse pack uses the BWM.Prune_Mart_Control table to store the data mart data prune values. By default, all hourly and daily data mart data older than 3 months is deleted when the process runs and all weekly and monthly data mart data older than 1 year is deleted.

This process runs automatically as the last step of the data mart ETL (ETL2) process.

The BWM.Prune_Mart_Log table keeps a history of the data mart data removal.

5.2.2.1 Pruning data mart data (table Prune_Mart_Control)

Data mart data is deleted from fact tables after it is either 90 days old for hourly or daily data or 1 year old for weekly and monthly data. This is based on the duration specified in the PMartC_Duration column of the BWM.Prune_Mart_Control table for each data mart table. You can modify the data mart prune values by running the following SQL statements, where *X* is a date duration whose format is *yyyymmdd* (for example: *X* = 00000108 for 0000 years, 01 month, 08 days).

Change hourly data mart prune values with:

Connect to TWH_MART

```
UPDATE BWM.Prune_Mart_Control
    SET PMartC_Duration = X
    WHERE Table_Name = 'BWM.F_TX_ND_HOUR'
```

Change daily data mart prune values with:

Connect to TWH_MART

```
UPDATE BWM.Prune_Mart_Control
    SET PMartC_Duration = X
    WHERE Table_Name = 'BWM.F_TX_ND_DAY'
```

Change weekly data mart prune values with:

Connect to TWH_MART

```
UPDATE BWM.Prune_Mart_Control
    SET PMartC_Duration = X
    WHERE Table_Name = 'BWM.F_TX_ND_WEEK'
```

Change monthly data mart prune values with:

Connect to TWH_MART

```
UPDATE BWM.Prune_Mart_Control
    SET PMartC_Duration = X
    WHERE Table_Name = 'BWM.F_TX_ND_MONTH'
```

See section 5.4, Other Maintenance Tools, for information on scripts that will set prune values for you.

5.3 Extraction control (table *Extract_Control*)

The extraction control table assists you in incrementally extracting data from a source database. For an example of incremental extraction, see the *Enabling an Application for Tivoli Data Warehouse* guide. Also, see section 7.5, Incremental Extraction, for additional information.

5.3.1 Measurement Data

The data extracted by the BWM_c10_src_to_cdw_Process is controlled by the values contained in the TWG.Extract_Control table in the Tivoli Data Warehouse. This table contains the sequence id for the first and last data rows that were extracted from the AggregateData table the last time the BWM_c10_src_to_cdw_Process ran. The BWM_c10_src_to_cdw_Process uses these values to extract only those data rows with sequence ids higher than the highest sequence id extracted during the previous run. This keeps the extract process from reprocessing prior events. Data from other Tivoli Monitoring for Transaction Performance source tables is extracted based on the AggregateData rows that correlate to those additional tables. Based on the data correlations new policy groups and agent groups will not be present in the central data warehouse until it's associated with aggregate data that is extracted.

This warehouse pack only uses the ExtCtl_From_IntSeq and ExtCtl_To_IntSeq columns to keep track of the rows that have been extracted from the source database. Initially the From and To columns are initialized to -1. This indicates to the extraction process to extract all data if the From is -1. The extract reads the From value and then finds the current maximum sequence ID of the AGGREGATEDATA table and stores this into the To column. When the data extraction from the AGGREGATEDATA table has successfully completed the Extract_Log table is updated with the From and To values used for the current extraction. When the Extract_Log is updated, a database trigger, internal to the Tivoli Data Warehouse, updates the Extract_Control ExtCtl_From_IntSeq column with the new value saved in the Extract_Control ExtCtl_To_IntSeq column. This prepares the table to be ready for the next extract process to run. For example, the first ETL run will read the From value from ExtCtl_From_IntSeq and get -1. These values are inserted into a new temporary table until the extract runs. Then the central data warehouse ETL process extracts the AGGREGATEDATA data and other required data. The central data warehouse ETL updates the ExtCtl_To_IntSeq to the maximum sequence ID found for the AGGREGATEDATA table that was stored in the temporary table. The last statement in the central data warehouse ETL updates the Extract_Log. This update causes the internal warehouse trigger to update the ExtCtl_From_IntSeq to the same value as ExtCtl_To_IntSeq preparing for the next extract process.

The following source tables are the Tivoli Monitoring for Transaction Performance source tables whose data is extracted by central data warehouse ETL (ETL1) into the central data warehouse:

EP, EPG, EpEGPInverse_EP, TRANSACTION, ARM_USER, APPLICATION, NODE, AGGREGATEDATA, RELATIONMAP, PT, THRESHOLD, TR, MANAGEMENTPOLICY, PolicyGroup, and Mpolicies_PGSP.

Only the AGGREGATEDATA table uses extract control because the data pulled from the other tables is used to further define the new measurement data being extracted from the AGGREGATEDATA table. The number and period shown before the AGGREGATEDATA names in the table below are used by Tivoli Data Warehouse, Version 1.2, to control the data extraction from multiple source databases.

Note: Multiple source databases are not supported by this warehouse pack at this time.

See section 5.4, Other Maintenance Tools, for information on scripts that will set extract control values for you.

ExtCtl_Source VARCHAR (120)	ExtCtl_Target VARCHAR (120)	ExtCtl_From_RawSeq CHAR (10)	ExtCtl_to_RawSeq CHAR (10)	ExtCtl_From_IntSeq BIGINT	ExtCtl_To_IntSeq BIGINT	ExtCtl_From_DtTm TIMESTAMP	ExtCtl_To_DtTm TIMESTAMP	Msrc_Corr_Cd CHAR (6)
I.AGGREGATEDATA	BWM.STAGE_AGGREG_DATA			20	20			BWM
TWG.MSMT	BWM.STG1_TX_ND_HR			203	203			BWM
TWG.MSMT	BWM.STG2_TX_ND_HR			203	203			BWM
BWM.STG_TX_ND_MET	BWM.T_TX_ND_METRIC			18	18			BWM
TWG.COMP	BWM.T_APP			45	45			BWM
TWG.COMP	BWM.T_HOST			67	67			BWM
TWG.COMP	BWM.T_TX			35	35			BWM
TWG.COMP	BWM.T_TX_ND			96	96			BWM
BWM.COMP_ATTR_LONG	BWM.COMP_ATTR_LONG			29	29			BWM
BWM.COMP_NAME_LONG	BWM.COMP_NAME_LONG			29	29			BWM

5.3.2 Data Mart Data

The data extracted by the BWM_m05_cdw_to_mart_Process is also controlled by the values contained in the TWG.Extract_Control table in the Tivoli Data Warehouse. This table contains the sequence id for the first and last data rows that were extracted the last time the BWM_m05_cdw_to_mart_Process ran. The BWM_m05_cdw_to_mart_Process uses these values to extract only those data rows with sequence ids higher than the highest sequence id extracted during the previous run. This keeps the extract process from reprocessing prior events. See section 5.4, Other Maintenance Tools, for information on scripts that will set extract control values for you.

The following source tables listed in the figure above are central data warehouse tables whose data is extracted by data mart ETL (ETL2) into the data mart tables:

TWG.Msmt, TWG.Comp, BWM.STG_TX_ND_MET, BWM.COMP_NAME_LONG, and BWM.COMP_ATTR_LONG.

5.4 Maintenance scripts

The scripts described below will help you list or change values in the Extract Control or Extract Log warehouse tables. Each script contains a statement to connect to the central data warehouse. The default value is TWH_CDW. If your warehouse pack configuration uses a different central data warehouse than TWH_CDW, modify the scripts to point to the central data warehouse name your warehouse pack uses.

5.4.1 Show_ExtCtl_Values.sql

The Show_ExtCtl_Values.sql script shows the Extract Control table values for this warehouse pack's tables that are extracted from using extract control.

Run this script from the DB2 Universal Database command line processor as a user that has permissions to view warehouse tables.

Example of how to run the script:

```
db2 -tvf Show_ExtCtl_Values.sql
```

5.4.2 Show_ExtLog_Values.sql

The Show_ExtLog_Values.sql script shows the Extract Log extraction windows each time an extract has been run.

Run this script from the DB2 Universal Database command line processor as a user that has permissions to view warehouse tables.

Example of how to run the script:

```
db2 -tvf Show_ExtLog_Values.sql
```

5.4.3 Show_Prune_Mart_Values.sql

The Show_Prune_Mart_Values.sql script shows the pruning values set for the data mart fact tables. These tables are the tables in the warehouse data mart database that hold application measurement data used for reporting purposes.

Run this script from the DB2 Universal Database command line processor as a user that has permissions to view warehouse tables.

Example of how to run the script:

```
db2 -tvf Show_Prune_Mart_Values.sql
```

5.4.4 Show_Prune_Msmt_Values.sql

The Show_Prune_Msmt_Values.sql script shows the pruning values set for the TWG.Msmt table in the central data warehouse. This is the table in the central data warehouse that holds application measurement data.

Run this script from the DB2 Universal Database command line processor as a user that has permissions to view warehouse tables.

Example of how to run the script:

```
db2 -tvf Show_Prune_Msmt_Values.sql
```

5.4.5 Reset_ETL1_extract_window.sql

The Reset_ETL1_extract_window.sql script resets the Extract Control window for the central data warehouse ETL extract process to have From and To values of -1. This causes the next central data warehouse ETL extract process to re-extract all data in the IBM Tivoli Monitoring for Transaction Performance database. Only use this if you know the data to extract from the IBM Tivoli Monitoring for Transaction Performance source database. The IBM Tivoli Monitoring for Transaction Performance source database does not duplicate measurement data to insert into the Tivoli Data Warehouse. Trying to insert duplicate measurement data into the Tivoli Data Warehouse will cause the extract process to fail.

You should use this script only to restart the Extract Control window for the BWM_c10_CDW_Process. If you want to reset the window to the last extract, use the Show_ExtLog_Values.sql script to show the extract_log values used for the last extract. Then make a copy of the Reset_ETL1_extract_window.sql script, giving it a new script name. Change the ExtCtl_To_IntSeq and ExtCtl_To_IntSeq values to be set in the new script from -1 to the values shown in the Extract Log output.

Run this script from the DB2 Universal Database command line processor as a user that has permissions to view warehouse tables:

```
db2 -tvf Reset_ETL1_extract_window.sql
```

5.4.6 Reset_ETL2_extract_window.sql

The Reset_ETL2_extract_window.sql script resets the Extract Control window for the data mart ETL extract process to have From and To values of -1. This causes the next data mart ETL extract process to re-extract all BWM measurement data in the Tivoli Data Warehouse database. Only use this if you know the data to be extracted from the Tivoli Data Warehouse database. The IBM Tivoli Monitoring for Transaction Performance source database does not duplicate measurement data to insert into the BWM Data mart database. Trying to insert duplicate data into the BWM Data mart database will cause the extract process to fail.

Use the following script only to restart the Extract Control window for the BWM_m05_cdw_to_mart_Process. If you want to reset the window to the last extract, use the Show_ExtLog_Values.sql script to show the extract_log values used for the last extract for the data mart ETL tables. Then make a copy of the Reset_ETL2_extract_window.sql script, giving it a new script name. Change the ExtCtl_To_IntSeq and ExtCtl_To_IntSeq values to be set in the new script from -1 to the values shown in the Extract Log output for the appropriate tables.

Run the following script from the DB2 command line processor as a DB2 user that has permissions to view warehouse tables:

```
db2 -tvf Reset_ETL2_extract_window.sql
```

5.4.7 Reset_ETL1_ETL2_extract_windows.sql

The Reset_ETL1_ETL2_extract_windows.sql script resets the Extract Control window for both the central data warehouse ETL and data mart ETL extract processes to have From and To values of -1. This causes the next central data warehouse ETL and data mart ETL extract processes to re-extract all data in the IBM Tivoli Monitoring for Transaction Performance database and Tivoli Data Warehouse for this warehouse pack, respectively. Only use this if you know the data to be extracted from the source database will not cause duplicate data inserted in the BWM Data mart database. Trying to insert duplicate data into the Tivoli Data Warehouse or BWM Data mart database will cause the extract process to fail.

Use the following script only to restart the Extract Control windows for both the BWM_c10_src_to_cdw_Process and BWM_m05_cdw_to_mart_Process. If you want to reset the window to the last extract values for each, use the Show_ExtLog_Values.sql script to show the extract_log values used for the last extract for the central data warehouse ETL and data mart ETL tables. Then make a copy of the Reset_ETL1_ETL2_extract_windows.sql script, giving it a new script name. Change the ExtCtl_To_IntSeq and ExtCtl_To_IntSeq values to be set in the new script from -1 to the values shown in the Extract Log output for the appropriate tables.

Run this script from the DB2 Universal Database command line processor as a user that has permissions to view warehouse tables.

```
db2 -tvf Reset_ETL1_ETL2_extract_windows.sql
```

5.4.8 Reset_ETL1_ETL2_prune_values_to_clear_data.sql

The Reset_ETL1_ETL2_prune_values_to_clear_data.sql script resets the pruning values for both the central data warehouse ETL (ETL1) and data mart ETL (ETL2) prune processes to values of -1. After

setting this value run the CDW_c05_Prune_and_Mark_Active process and the BWM_m05_s050_mart_prune step of the BWM_m05_cdw_to_mart_Process. This will cause the CDW BWM measurement data and BWM data mart data to be removed. Only use this if you know the data to be removed is no longer needed in the Tivoli Data Warehouse because the only way to replace it is by re-extracting it from the Tivoli Monitoring for Transaction Performance source database, if the data still exists there.

Run this script from the DB2 Universal Database command line processor as a user that has permissions to view warehouse tables.

Example of how to run the script:

```
db2 -tvf Reset_ETL1_ETL2_prune_values_to_clear_data.sql
```

5.4.9 Reset_ETL1_ETL2_prune_values_to_defaults.sql

The Reset_ETL1_ETL2_prune_values_to_defaults.sql script resets the pruning values for both the central data warehouse ETL (ETL1) and data mart ETL (ETL2) prune processes to the values they were installed with. See section 5.2, Deleting data, for more information on what prune values are set at install time. If you want to set the prune timeframe to alternate values then make a copy of the script and change the numbers after 'PMsmtC_AGE_In_Days =' or 'PMartC_Duration =' on the 'SET' statements to the desired value.

Run this script from the DB2 Universal Database command line processor as a user that has permissions to view warehouse tables.

Example of how to run the script:

```
db2 -tvf Reset_ETL1_ETL2_prune_values_to_defaults.sql
```

5.4.10 Create_Cross_App_report_views.sql

The Create_Cross_App_report_views.sql script creates the views necessary to run the cross-application report, *'Resource usage, response time, and execution load for transaction <transaction>'* report. The script should only be run under the following circumstances:

- One time to create views required to run the Tivoli Monitoring for Transaction Performance or IBM Tivoli Monitoring cross application report
- After the IBM Tivoli Monitoring AMY warehouse pack has been installed and is extracting data
- When the IBM Tivoli Monitoring AMY warehouse pack's data mart tables exist in the same warehouse data mart database as the Tivoli Monitoring for Transaction Performance warehouse pack's data mart tables

Run this script from the DB2 Universal Database command line processor as a user that has permissions to view warehouse tables.

```
db2 -tvf Reset_ETL1_ETL2_prune_values_to_defaults.sql
```

5.4.11 Clean_and_Reset_BWM_warehouse_Delete_tables.sql

The Clean_and_Reset_BWM_warehouse_Delete_tables.sql script will remove the ETL tables as if this warehouse pack were just installed. This script is the first of a two-step process to remove warehouse data for the BWM warehouse pack and should only be run when it is desired that all the data for the BWM warehouse pack be removed from both the central data warehouse and MART databases.

This script must be run when the Tivoli Monitoring for Transaction Performance source database is changed to point to a different source database. The reason for this is that the sequence id's that are used in the source database could be the same when the database source is changed. Since these values are used in the ETL process as well, finding the same sequence id's in the new source database for different data records can cause data corruption and invalid relations in the central data warehouse. The script also

can be useful when the central data warehouse ETL has been running in a test environment in preparation for a production environment, you can run this script to remove all the test data and reset the extract control and prune values.

This first script drops the tables belonging to the BWM warehouse pack. The next script, `Clean_and_Reset_BWM_warehouse_Create_tables.sql`, should be run after this script to remove data from the central data warehouse tables and recreate the necessary warehouse pack tables. All central data warehouse measurements, components, component attributes, and component relationships will be removed with the exception of SHARED data. The only data that will be left in the CDW from this warehouse pack will be any SHARED components and their relationships. For this warehouse pack the only SHARED components are the J2EE* components and the IP_HOST component. Any relations for a SHARED component that had a relation with BWM component will also be removed. It also clears out any data that might exist in a staging table used by the central data warehouse ETL process. It resets the Extract Control starting timestamp (`ExtCtl_Strt_DtTm`) back to '1970-01-01-00.00.00.000000' and the prune values will be reset to their installation default values.

Limitations:

This script may run with errors because some tables may not currently exist that are being removed. Ignore errors from this script indicating particular tables did not exist.

This script should only be run when no warehouse ETL's are running to avoid database lock contention and problems with the ETL. The transactions logs must be sized large enough to handle the data deletion or the data could be rolled back after running for long periods of time. To improve performance after removing many rows it is recommended to update the database statistics.

Run this script from the DB2 Universal Database command line processor as a user that has permissions to view warehouse tables.

```
db2 -tvf Clean_and_Reset_BWM_warehouse_Delete_tables.sql
```

5.4.12 Clean_and_Reset_BWM_warehouse_Create_tables.sql

The `Clean_and_Reset_BWM_warehouse_Create_tables.sql` script will remove the central data warehouse data for this warehouse pack, with the exception of the SHARED components, and then recreate all tables, indexes, triggers, sequences and views required by this warehouse pack that were removed by the first script, `Clean_and_Reset_BWM_warehouse_Delete_tables.sql`. This script is the second of a two-step process to remove warehouse data for the BWM warehouse pack.

This script must be run when the Tivoli Monitoring for Transaction Performance source database is changed to point to a different source database. The reason for this is that the sequence id's that are used in the source database could be the same when the database source is changed. Since this values are used in the ETL process as well, finding the same sequence id's in the new source database for different data records can cause data corruption and invalid relations in the central data warehouse. The script also can be useful when the central data warehouse ETL has been running in a test environment in preparation for a production environment, you can run this script to remove all the test data and reset the extract control and prune values.

All central data warehouse measurements, components, component attributes, and component relationships will be removed with the exception of SHARED data. The only data that will be left in the CDW from this warehouse pack will be any SHARED components and their relationships. For this warehouse pack the only SHARED components are the J2EE* components and the IP_HOST component. Any relations for a SHARED component that had a relation with BWM component will also be removed. It also clears out any data that might exist in a staging table used by the central data warehouse ETL process. It resets the Extract Control starting timestamp (`ExtCtl_Strt_DtTm`) back to '1970-01-01-00.00.00.000000' and the prune values will be reset to their installation default values.

Limitations:

This script may run with errors because some tables may not currently exist that are being removed. Ignore errors from this script indicating particular tables did not exist.

This script should only be run when no warehouse ETL's are running to avoid database lock contention and problems with the ETL. The transactions logs must be sized large enough to handle the data deletion or the data could be rolled back after running for long periods of time. To improve performance after removing many rows it is recommended to update the database statistics.

Run this script from the DB2 Universal Database command line processor as a user that has permissions to view warehouse tables. This script will stop running on the first error encountered if run with

```
db2 -stvf Clean_and_Reset_BWM_warehouse_Create_tables.sql
```

5.5 Problem determination

If a step in one of the extract processes fails, review the logs in the \$TIVOLI_COMMON_DIR/cdw/logs/etl directory, correct the problem, and rerun the step that failed.

For common problems and solutions, see *the Installing and Configuring Tivoli Data Warehouse Guide* and *the Enabling an Application for Tivoli Data Warehouse Guide*.

5.5.1 Crystal error for report Resource Usage, Response Time and Execution Load For Transaction <transaction>

When the IBM Tivoli Monitoring AMY pack is not installed or the Create_Cross_App_report_views.sql script has not been run the following error message will be seen.

“There was an error retrieving data from the server: Failed to open a rowset. File #####.rpt.”

If the IBM Tivoli Monitoring AMY warehouse pack mart tables don't exist in the same TDW_MART database as the BWM warehouse pack or the Create_Cross_App_report_views.sql script was not run to create the views required by the cross-application report Resource Usage, Response Time and Execution Load For Transaction <transaction> the report cannot find the data it requires to work properly.

5.5.2 Sample ETL Log Informational messages

The following is an example of the messages that can be seen in the logs of the ETL steps. These messages can be informational, warning or error messages.

***** WARNING ***** Warning Text: The host, prague.tivlab.austin.ibm.com, does not belong to an agent group so it will not be listed in any reports by agent group.

Recovery: No manual recovery steps needed but research why this host does not belong to an agent group as it should.

Calculation: checking to see if any hosts do not belong to an agent group

Value: Host name: prague.tivlab.austin.ibm.com

Allowable Low Value: N/A

Allowable High Value: N/A

```
WARNING_LABEL    WARNING_TEXT
WARNING_RECOVERY
CALCULATION
ACTUAL_VALUE
LOW_VALUE
HIGH_VALUE
```

```
-----
-----
```

[illegible]

***** WARNING *****

Warning Text: The policy name,

Alargenapolicyxx
xx
xx
xxNOTENDL
ARGE, will be truncated to 254 characters to fit in the warehouse TWG.Comp table comp nm column.

Recovery: No manual recovery steps needed but you may want to be aware that the name has exceeded the length allowed in the warehouse and may affect having unique management policy names.

Calculation: checking to see if any policy names will be truncated

Value: Policy name:

Alargenampolicydd
 dddddddddddddddddddxxxddddddddddddddddddddddddddddddxxxdddddddddddddd
 dddddddddddddddxxxddddddddddddddddddddddddddddddNOTENDL
 ARGEE

Allowable Low Value: N/A

Allowable High Value: N/A

6 ETL processes

The warehouse pack has the following processes:

- BWM_c05_Upgrade_Process
- BWM_c10_src_to_cdw_Process
- BWM_m05_cdw_to_mart_Process

6.1 *BWM_c05_Upgrade_Process*

This process upgrades a Tivoli Monitoring for Transaction Performance warehouse pack version 5.2 central data warehouse and data mart data to the 5.3 data needed for this version of the warehouse pack. It should be run only one time. It can be rerun, though, if there are any failures when it runs. If a failure occurs look at the log file to see what happened, correct the problem and then rerun the script. Once the script has run all the way through successfully it should not be run again.

TWH_CDW SCHEMA CHANGES:

In the central data warehouse, the following schema changes are made during the upgrade:

- A BWM.EXCEPTION_LOG table is created.

TWH_MART SCHEMA CHANGES:

In the mart, the following schema changes are made during the upgrade:

- A BWM.EXCEPTION_LOG table is created.
- A BWM.D_TX_ND_POL_GRPS table is created.
- A BWM.D_HOST_AGENT_GRPS table is created.

Once the 5200 warehouse pack is upgraded to 5300, the BWM_c05_Upgrade_Process should be run. It should only need to be run once, but can be rerun, if necessary, due to any failures or corrections that may need to be made.

If a step in the upgrade process fails, review the logs in the \$TIVOLI_COMMON_DIR/cdw/logs/etl directory, correct the problem, and rerun the upgrade process.

Example:

c:\Program Files\IBM\tivoli\common\cdw\logs\etl\BWM_c05_s080_cdw_long_names.log

To check for any exceptions logged during the BWM_c05_Upgrade_Process, use the following SQL statement on the BWM.EXCEPTION_LOG table in the central data warehouse.

```
“SELECT * FROM BWM.EXCEPTION_LOG”
```

The table entries with the latest timestamp in the Reported_DtTm field will be from the last run of the process. If, for any reason, the process did not complete, then the entries may have a Reported_DtTm timestamp value of ‘9999-01-01-00.00.00.000000’. This is because the last step in the process resets the Reported_DtTm value to the same timestamp in order to correlate the exceptions logged during the process. In this case, look at the Error_DtTm field to determine which entries correspond to the timeframe in which you ran the last process.

Once the script has run all the way through successfully, you should not need to run it again.

TWH_CDW DATA CHANGES:

In the central data warehouse, the process will make the following data changes:

- For every BWM_TX_NODE attribute of type BWM_MGMT_POLICY, a new component of type BWM_MGMT_POLICY will be created.
- For every BWM_MGMT_POLICY component created, a relationship to its corresponding BWM_TX_NODE will be created with relationship type DESCR1.
- For every BWM_MGMT_POLICY component created, a new BWM_MGMT_POL_GRP component will be created with the same name as the BWM_MGMT_POLICY component.
- For each BWM_MGMT_POLICY component created, a relationship to its corresponding BWM_MGMT_POL_GRP will be created with relationship type LCONT.
- For every BWM_AGENT_GRP with a BWM_HOST or IP_HOST relationship defined in the Source Database, a new component of type BWM_AGENT_GRP will be created.
- For every BWM_AGENT_GRP component created, a relationship to its corresponding BWM_HOST or IP_HOST will be created with relationship type LCONT.
- For every BWM_HOST or IP_HOST component that does not have an agent group relationship after the Source Database new agent group information has been processed, a BWM_AGENT_GRP component will be created with the same name as the BWM_HOST or IP_HOST component. A relationship to its corresponding BWM_HOST or IP_HOST will be created with relationship type LCONT.
- For every J2EE_SERVER component, a relationship to its corresponding BWM_HOST or IP_HOST components will be created with relationship type RUNSON. The following entries are deleted from the TWG.COMP table:
comptyp_cd='J2EE_SERVER' and compnm='.*'
comptyp_cd='J2EE_NODE' and compnm='N/A'
 'J2EE_DOMAIN' and compnm='N/A'
 'J2EE_CELL' and compnm='N/A'
- For every entry in the BWM.COMP_NAME_LONG table, a corresponding entry will be created in the new TWG.COMP_EXT table. Any name longer than 3500 characters will be truncated to 3500 characters in the new TWG.COMP_EXT table. The COMP_NAME_LONG table will not be removed after upgrade. This enables the upgrade to be rerun and non-destructive. The customer may delete the table when they are satisfied the upgrade is complete, should they want to reclaim its space. It will also be removed in the subsequent warehouse pack.

TWH_MART DATA CHANGES:

In the mart, the process will make the following changes:

- For each new BWM_MGMT_POL_GRP component created in the central data warehouse, a relationship will be created in the data mart in the new BWM.D_TX_ND_POL_GRP table to its corresponding policy entry in the BWM.D_TX_ND table.
- For each new BWM_AGENT_GRP component created in the central data warehouse, a relationship will be created in the data mart in the new BWM.D_HOST_AGENT_GRP table to its corresponding policy entry in the BWM.D_HOST table.

This process has the following steps:

- **BWM_c05_s010_upg_cdw_exception_pre_check**
This step runs exception checks for any items of interest prior to running the upgrade and logs items of note to the exception log table.

The SOURCE and TARGET database for this step is TWH_CDW.

Example:

- a. Has the 5300 upgrade or 5300 central data warehouse ETL (ETL1) already been run?
- b. How many components of type 'J2EE__SERVER' do not have a 'BWM_HOST' relationship of type USES (needed to map J2EE_SERVER to it's HOST)?
- c. How many components of type 'BWM_TX_NODE' do not have an IP_HOST or BWM_HOST relationship of type RUNSON (needed to map J2EE_SERVER to its HOST)?
- d. How many component names of length > 3500 exist (these will be truncated to 3500 characters during upgrade)?

- **BWM_c05_s020_upg_cdw_mgmtpolicy**

This step inserts all BWM_MGMT_POLICY attributes as new components into the TWG.COMP table. It then adds to the TWG.COMPRELN table relationships for the new BWM_MGMT_POLICY components to their BWM_TX_NODE components of type DISTRI.

The SOURCE and TARGET database for this step is TWH_CDW.

The following exceptions may be logged:

An exception will be logged for the number of distinct policy names in the TWG.COMPATTR table of type BWM_MGMT_POLICY that do not have a component entry in the TWG.COMP table of type BWM_MGMT_POLICY.

An exception will be logged for the number of policies in the TWG.COMPATTR table of type BWM_MGMT_POLICY that do not have relationship entries in the TWG.COMPRELN table of type DESCR.

- **BWM_c05_s030_upg_cdw_mgmtpolicygroup**

This step adds BWM_MGMT_POL_GRP new data as new component instances into the TWG.COMP table. It then adds to the TWG.COMPRELN table relationships for the new BWM_MGMT_POL_GRP components to their BWM_MGMT_POLICY components of type LCONT.

The SOURCE and TARGET databases for this step are TWH_CDW.

The following exceptions may be logged:

An exception will be logged for the number of distinct policy names in the TWG.COMP table of type BWM_MGMT_POLICY that do not have a matching component entry with the same name in the TWG.COMP table of type BWM_MGMT_POL_GRP.

An exception will be logged for the number of policies in the TWG.COMP table of type BWM_MGMT_POLICY that do not have relationship entries in the TWG.COMPRELN table to their BWM_MGMT_POL_GRP component of type LCONT.

- **BWM_c05_s040_upg_mart_d_tx_nd_pol_grps**

This step inserts rows into the new BWM.D_TX_ND_POL_GRPS table for each entry in the BWM.D_TX_ND table that has a matching relationship in the TWG.COMPRELN table for BWM_MGMT_POLICY components to BWM_MGMT_POL_GRP components.

The SOURCE database for this step is TWH_CDW.

The TARGET database for this step is TWH_MART.

The following exception may be logged:

An exception will be logged for the number of distinct transaction node ids in the BWM.D_TX_ND table that do not have a matching new entry with the same id in the TWG.D_TX_ND_POL_GRPS table.

- **BWM_c05_s050_upg_cdw_agentgroup**

This step adds BWM_AGENT_GRP new data as new component instances into the TWG.COMP table. It then adds to the TWG.COMPRELN table a relationship for the new BWM_AGENT_GRP components to their BWM_HOST or IP_HOST components of type LCONT.

The SOURCE database for this step is TMTP_DB_SRC.

The TARGET database for this step is TWH_CDW.

The following exceptions may be logged:

An exception will be logged for the number of distinct agent group names in the Source database that do not have a matching component entry with the same name in the TWG.COMP table of type BWM_AGENT_GRP.

An exception will be logged for the number of distinct agent names in the Source database of type BWM_HOST or IP_HOST matched with an agent group that do not have matching component relationship entries in the TWG.COMPRELN table of type LCONT.

An exception will be logged for the number of distinct agent names in the TWG.COMP table of type BWM_HOST or IP_HOST that do not have a matching component entry with the same name in the TWG.COMP table of type BWM_AGENT_GRP.

An exception will be logged for the number of distinct agents in the TWG.COMP table of type BWM_HOST or IP_HOST that do not have relationship entries in the TWG.COMPRELN table to their BWM_AGENT_GRP component of type LCONT.

- **BWM_c05_s060_upg_mart_d_host_agent_grps**

This step inserts rows into the new BWM.D_HOST_AGENT_GRPS table for each entry in the BWM.D_HOST table that has a matching relationship in the TWG.COMPRELN table for BWM_HOST and/or IP_HOST components to BWM_AGENT_GRP components.

The SOURCE database for this step is TWH_CDW.

The TARGET database for this step is TWH_MART.

The following exception may be logged:

An exception will be logged for the number of distinct host ids in the BWM.D_HOST table that do not have a matching new entry with the same id in the TWG.D_HOST_AGENT_GRPS table.

- **BWM_c05_s070_upg_cdw_j2eehost**

This step adds to the TWG.COMPRELN table relationships for J2EE_SERVER components to their BWM_HOST and/or IP_HOST components of type RUNSON.

The SOURCE and TARGET databases for this step are TWH_CDW.

The following exception may be logged:

An exception will be logged for the number of distinct components in the TWG.COMP table of type J2EE_SERVER that do not have relationship entries in the TWG.COMPRELN table to their BWM_HOST or IP_HOST component of type LCONT.

- **BWM_c05_s080_upg_cdw_long_names**

This step inserts rows into the new TWG.COMP_EXT table for each entry in the BWM.COMP_NAME_LONG table, truncating the name to length 3500 where it is longer than this.

The SOURCE and TARGET database for this step is TWH_CDW.

The following exception may be logged:

An exception will be logged for the number of distinct component names in the BWM.COMP_NAME_LONG table that do not have a corresponding new entry that matches in the TWG.COMP_EXT table.

6.2 BWM_c10_src_to_cdw_Process

This process extracts data from the IBM Tivoli Monitoring for Transaction Performance source database, transforms it, and loads it into the central data warehouse database. This process should be run once a day before the data mart ETL is run. Measurement data is inserted into the Tivoli Data Warehouse when there are new measurements. The data that goes with a measurement such as node, host, application is only entered in the Tivoli Data Warehouse as new data when a measurement that uses those values is inserted into the Tivoli Data Warehouse.

Note: If you rerun this step before the data from the previous run has been loaded into the central data warehouse (for example, if the previous run failed but was not rerun from the failing step), the staging tables are dropped and emptied in the BWM_c10_s010_pre_extract step and any data in the dropped staging tables is lost.

This process has the following steps:

- **BWM_c10_s010_src_pre_extract**

This step drops and recreates all the staging tables that were used in a previous run of the ETL. These staging tables are populated during the extract step of the ETL.

- **BWM_c10_s020_src_extract**

This step performs the extraction of new data from the IBM Tivoli Monitoring for Transaction Performance source database into the staging tables in the central data warehouse database. Once the data has been extracted, the TWG.Extract_Control table is updated with the highest sequence ID extracted and the TWG.Extract_Log is updated with the range of sequence IDs that were extracted.

- **BWM_c10_s030_src_load**

This step transforms the IBM Tivoli Monitoring for Transaction Performance data in the BWM staging tables into the desired components, attributes, relationships, and measurements and inserts them into the TWG.Comp, TWG.CompAttr, TWG.CompReln and TWG.Msmt tables.

6.3 *BWM_m05_cdw_to_mart_Process*

This process extracts data from the central data warehouse database and transforms and loads it into the IBM Tivoli Monitoring for Transaction Performance data mart tables. This process is run automatically once a day after the central data warehouse ETL runs successfully.

This process has the following steps:

- **BWM_m05_s005_mart_prepare_extract**

This step creates and populates the central data warehouse staging tables for this warehouse pack with the data that will be extracted to the data mart database. This preparatory step helps with the performance of the data mart ETL.

- **BWM_m05_s010_mart_pre_extract**

This step clears the staging fact tables used in the extraction of data for the data mart tables.

- **BWM_m05_s020_mart_extract**

This step extracts the data from the central data warehouse database in to fill in the dimension translation tables and hourly staging fact tables in the data mart database with any new data.

- **BWM_m05_s030_mart_load**

This step loads the data from data mart staging tables into the actual data mart tables.

- **BWM_m05_s040_mart_rollup**

This step aggregates the hourly fact data into the daily, weekly, and monthly fact tables.

Once the Hourly fact tables have been populated by the BWM_m05_s030_mart_load step, the rollup step populates the daily, weekly, and monthly fact tables in the data mart based on the data in the staging fact tables. The staging fact tables only contain the current day's data.

- **BWM_m05_s050_mart_prune**

This step prunes the hourly and daily fact tables of data older than 3 months. Weekly and monthly fact tables are pruned of data that is more than a year old.

The 3-month duration value is a parameter that is set into the TWG.Prune_Msmt_Control table. The duration value is based on the format 'yyyymmdd', so an entry of 300 indicates 3 months of data to be pruned.

It is recommended that the data mart ETL process run once a day. The prune step is the last step of the data mart ETL process. In this way the prune step does not have to be scheduled separately.

7 Central data warehouse information

Before reading this section, read about the generic schema for the central data warehouse, which is described in *Enabling an Application for Tivoli Data Warehouse*. That document defines the content of each table and explains the relationships between the tables in this document.

This section provides an example of how information about IBM Tivoli Monitoring for Transaction Performance data is stored in Tivoli Data Warehouse.

This section about the information in the central data warehouse is intended primarily for report designers and warehouse pack creators. For information about reports, see “Reports” on page 21.

Shaded columns in the following tables are translated. These are also marked with an asterisk (*) after the column name.

7.1 Component configuration

The following sections describe the component configuration.

7.1.1 Component type (table CompTyp)

CompTyp_Cd CHAR (17)	CompTyp_Parent_Cd CHAR (17)	CompTyp_Nm * VARCHAR (120)	CompTyp_Strt_DtTm TIMESTAMP	CompTyp_End_DtTm TIMESTAMP	MSrc_Corr_Cd CHAR (6)
IP_HOST	NULL	IP Host	2002-06-30-12.00.00.000000	9999-01-01-12.00.00.000000	MODEL1
IP_INTERFACE	NULL	IP Interface	2002-06-30-12.00.00.000000	9999-01-01-12.00.00.000000	MODEL1
BWM_HOST	NULL	Transaction Host	2002-06-30-12.00.00.000000	9999-01-01-12.00.00.000000	BWM
J2EE_SERVER	NULL	J2EE Server	2002-06-30-12.00.00.000000	9999-01-01-12.00.00.000000	MODEL1
J2EE_NODE	NULL	J2EE Node	2002-06-30-12.00.00.000000	9999-01-01-12.00.00.000000	MODEL1
J2EE_DOMAIN	NULL	J2EE Domain	2002-06-30-12.00.00.000000	9999-01-01-12.00.00.000000	MODEL1
J2EE_CELL	NULL	J2EE Cell	2002-06-30-12.00.00.000000	9999-01-01-12.00.00.000000	MODEL1
BWM_TX_NODE	NULL	Transaction Node	2002-06-30-12.00.00.000000	9999-01-01-12.00.00.000000	BWM

CompTyp_Cd CHAR (17)	CompTyp_Parent_Cd CHAR (17)	CompTyp_Nm * VARCHAR (120)	CompTyp_Strt_DtTm TIMESTAMP	CompTyp_End_DtTm TIMESTAMP	MSrc_Corr_Cd CHAR (6)
BWM_TRANSACTION	NULL	Transaction	2002-06-30-12.00.00.000000	9999-01-01-12.00.00.000000	BWM
BWM_PROBE	NULL	Monitoring Probe	2002-06-30-12.00.00.000000	9999-01-01-12.00.00.000000	BWM
BWM_AGENT_GRP	NULL	Agent Group	2002-06-30-12.00.00.000000	9999-01-01-12.00.00.000000	BWM
BWM_MGMT_POLICY	NULL	Management Policy	2002-06-30-12.00.00.000000	9999-01-01-12.00.00.000000	BWM
BWM_MGMT_POL_GRP	NULL	Management Policy Group	2002-06-30-12.00.00.000000	9999-01-01-12.00.00.000000	BWM

7.1.2 Component extension (table Comp_ext)

Comp_ID INTEGER	Comp_Long_Nm VARCHAR (3500)
7	http:// www-132.ibm.com:80/ webapp/wcs/stores/servlet/PromotionDisplay?promoId=10922&catalogId=-840&storeId=1&langId=-1&color=default&userid=tomking&convert=no&tint=azul&Default?Values=white&nextSite?Next=tomorrow&background=green&Programs_view=allsitesavailable&endtime=quit

The Comp_Ext table is used when the data to fit in the Comp table's Comp_Nm column is too long to fit into 254 characters. When the Comp_Nm is greater than 254 characters the value is truncated to 244 characters, an underscore is appended and then the value placed in the Comp_Corr_Val is appended to the end and this becomes the Comp_Nm. The full Comp_Nm value is then inserted into the Comp_ext table.

7.1.3 Component (table Comp)

Comp_ID INTEGER	CompTyp_Cd CHAR (17)	Centr_Cd CHAR (6)	Cust_ID INTEGER	Comp_Corr_ID INTEGER	Comp_Nm VARCHAR (254)	Comp_Corr_Val VARCHAR (254)	Comp_Strt_DfTm TIMESTAMP	Comp_End_DfTm TIMESTAMP	Comp_Ds VARCHAR (254)	Msrc_Corr_Cd CHAR (6)
1	BWM_HOST	CDW	1		host1		2002-06-30- 12.00.00.000000	9999-01-01- 12.00.00.000000	Host used for Berny's Tivoli Monitoring for Transaction Performance data	BWM
2	IP_HOST	CDW	1		host2.ibm.com		2002-06-30- 12.00.00.000000	9999-01-01- 12.00.00.000000	Tivoli Monitoring for Transaction Performance server machine for SW area	SHARED
3	BWM_PROBE	CDW	1	177	QoS	host1	2002-06-30- 12.00.00.000000	9999-01-01- 12.00.00.000000		BWM
4	BWM_TRANSACTION	CDW	1	264	http://www.ibm.com/ *		2002-06-30- 12.00.00.000000	9999-01-01- 12.00.00.000000		BWM
5	BWM_TRANSACTION	CDW	1	356	http://www.ibm.com/hr/index.html		2002-06-30- 12.00.00.000000	9999-01-01- 12.00.00.000000		BWM
6	BWM_TRANSACTION	CDW	1	864	Session.create()		2002-06-30- 12.00.00.000000	9999-01-01- 12.00.00.000000		BWM
7	BWM_TRANSACTION	CDW	1	753	http:// www-132.ibm.com:8		2002-06-30- 12.00.00.000000	9999-01-01- 12.00.00.000000		BWM

Comp_ID INTEGER	CompTyp_Cd CHAR (17)	Centr_Cd CHAR (6)	Cust_ID INTEGER	Comp_Corr_ID INTEGER	Comp_Nm VARCHAR (254)	Comp_Corr_Val VARCHAR (254)	Comp_Strt_DtTm TIMESTAMP	Comp_End_DtTm TIMESTAMP	Comp_Ds VARCHAR (254)	Msrc_Corr_Cd CHAR (6)
					0/ webapp/wcs/sto res/servlet/Pro motionDisplay? promoId=10922 &catalogId=- 840&storeId=1 &langId=- 1_753		12.00.00.000000	12.00.00.000000		
8	BWM_TX_NODE	CDW	1	544	com.ibm.petstor e. Session.create() _543	host2.ibm.com@1 3!24	2002-06-30- 12.00.00.000000	9999-01-01- 12.00.00.000000		BWM
9	BWM_TX_NODE	CDW	1	290	http://www.ibm .com/hr/index.h tml_642	host2.ibm.com@4 3!34	2002-06-30- 12.00.00.000000	9999-01-01- 12.00.00.000000		BWM
10	BWM_TX_NODE	CDW	1	96	http:// www- 132.ibm.com:8 0/ webapp/wcs/sto res/servlet/Pro motionDisplay? promoId=10922 &catalogId=- 840&storeId=1 &langId=- 1_6953	host2.ibm.com@4 5!48	2002-06-30- 12.00.00.000000	9999-01-01- 12.00.00.000000		BWM
11	J2EE_SERVER	CDW	1	513	Server1	Websphere!!5.0	2002-06-30- 12.00.00.000000	9999-01-01- 12.00.00.000000		SHARED
12	J2EE_SERVER	CDW	1	673	Server2	Websphere!!5.0	2002-06-30- 12.00.00.000000	9999-01-01- 12.00.00.000000		SHARED

Comp_ID INTEGER	CompTyp_Cd CHAR (17)	Centr_Cd CHAR (6)	Cust_ID INTEGER	Comp_Corr_ID INTEGER	Comp_Nm VARCHAR (254)	Comp_Corr_Val VARCHAR (254)	Comp_Strt_DtTm TIMESTAMP	Comp_End_DtTm TIMESTAMP	Comp_Ds VARCHAR (254)	Msrc_Corr_Cd CHAR (6)
							12.00.00.000000	12.00.00.000000		
13	J2EE_NODE	CDW	1	942	Peace		2002-06-30- 12.00.00.000000	9999-01-01- 12.00.00.000000		SHARED
14	J2EE_NODE	CDW	1	654	hope		2002-06-30- 12.00.00.000000	9999-01-01- 12.00.00.000000		SHARED
15	BWM_TX_NODE	CDW	1	365	http://www.ibm.com/*_832	host2.ibm.com@47!49	2002-06-30- 12.00.00.000000	9999-01-01- 12.00.00.000000		BWM
16	J2EE_DOMAIN	CDW	1	164	jdbc:db2:was40:2003.1.4.15.51.4.539		2002-06-30- 12.00.00.000000	9999-01-01- 12.00.00.000000		SHARED
17	J2EE_CELL	CDW	1	270	CELL1		2002-06-30- 12.00.00.000000	9999-01-01- 12.00.00.000000		SHARED
18	IP_HOST	CDW	1		Stewart1.ibm.com		2002-06-30- 12.00.00.000000	9999-01-01- 12.00.00.000000	NE J2EE Transaction Server	SHARED
19	BWM_TRANSACTION	CDW	1	349	http://www-132.ibm.com/webapp/wcs/store/servlet/PromotionDisplay?promoName=526372&storeId=1&catalogId=-840&langId=-1&dualCurrId=73		2002-06-30- 12.00.00.000000	9999-01-01- 12.00.00.000000		BWM

Comp_ID INTEGER	CompTyp_Cd CHAR (17)	Centr_Cd CHAR (6)	Cust_ID INTEGER	Comp_Corr_ID INTEGER	Comp_Nm VARCHAR (254)	Comp_Corr_Val VARCHAR (254)	Comp_Strt_DtTm TIMESTAMP	Comp_End_DtTm TIMESTAMP	Comp_Ds VARCHAR (254)	Msrc_Corr_Cd CHAR (6)
20	BWM_TX_NODE	CDW	1	287	http://www-132.ibm.com/webapp/wcs/stores/servlet/PromotionDisplay?promoName=526372&storeId=1&catalogId=840&langId=1&dualCurrId=73_1546	host2.ibm.com@52!61	2002-06-30-12.00.00.000000	9999-01-01-12.00.00.000000		BWM
21	BWM_TX_NODE	CDW	1	53	http://www.ibm.com/us/_5432	host2.ibm.com@60!65	2002-06-30-12.00.00.000000	9999-01-01-12.00.00.000000		BWM
22	BWM_TRANSACTION	CDW	1	740	http://www.ibm.com/us/securit y		2002-06-30-12.00.00.000000	9999-01-01-12.00.00.000000		BWM
23	BWM_MGMT_POLICY	CDW	1		Policy1		2002-06-30-12.00.00.000000	9999-01-01-12.00.00.000000		BWM
24	BWM_MGMT_POLICY	CDW	1		Policy9		2002-06-30-12.00.00.000000	9999-01-01-12.00.00.000000		BWM
25	BWM_MGMT_POLICY	CDW	1		EJBPolicy		2002-06-30-12.00.00.000000	9999-01-01-12.00.00.000000		BWM
26	BWM_AGENT_GRP	CDW	1		East_Coast		2002-06-30-12.00.00.000000	9999-01-01-12.00.00.000000		BWM
27	BWM_MGMT_POL_GRP	CDW	1		EJBPolicy_Grp		2002-06-30-12.00.00.000000	9999-01-01-12.00.00.000000		BWM

Comp_ID INTEGER	CompTyp_Cd CHAR (17)	Centr_Cd CHAR (6)	Cust_ID INTEGER	Comp_Corr_ID INTEGER	Comp_Nm VARCHAR (254)	Comp_Corr_Val VARCHAR (254)	Comp_Strt_DtTm TIMESTAMP	Comp_End_DtTm TIMESTAMP	Comp_Ds VARCHAR (254)	Msrc_Corr_Cd CHAR (6)
28	BWM_MGMT_POL_GRP	CDW	1		Policies1_to_9		2002-06-30- 12.00.00.000000	9999-01-01- 12.00.00.000000		BWM

The comp_ds component description values may be truncated when the description column in the source database is greater than 254 characters.

BWM_MGMT_POL_GRP and BWM_MGMT_POLICY are examples of two components whose descriptions may be truncated. The comp_nm field for the BWM_TRANSACTION component may be truncated if the transaction name is greater than 3500 characters, the length of the comp_long_nm column in the Comp_Ext table.

****Note:** If a transaction name exceeds 240 characters, the corresponding transaction name is truncated and the full transaction name is stored in the BWM.COMP_NAME_LONG table.

7.1.4 Component relationship type (table RelnTyp)

RelnTyp_Cd CHAR (6)	RelnTyp_Nm * VARCHAR (120)	Msrc_Corr_Cd CHAR (6)
PCHILD	Parent Child Relation	MODEL1
USES	Uses Relation	MODEL1
RUNSON	Runs on Relation	MODEL1
INVOKE	Invoke Relation	MODEL1
INSTON	Installed on Relation	MODEL1
LCONT	Logical Containment Relation	MODEL1
DESCRI	Described Monitoring Of Relation	MODEL1
* This column is translated		

7.1.5 Component relationship rule (table ReInRul)

CompTyp_Source_Cd CHAR (17)	CompTyp_Target_Cd CHAR (17)	ReInTyp_Cd CHAR (6)	ReInRul_Strt_DtTm TIMESTAMP	ReInRul_End_DtTm TIMESTAMP
BWM_PROBE	BWM_TRANSACTION	PCHILD	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000
BWM_PROBE	BWM_TX_NODE	INVOKE	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000
BWM_TX_NODE	BWM_HOST	RUNSON	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000
BWM_TX_NODE	J2EE_SERVER	USES	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000
BWM_TX_NODE	BWM_TRANSACTION	INSTOF	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000
BWM_PROBE	IP_INTERFACE	RUNSON	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000
BWM_PROBE	BWM_HOST	RUNSON	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000
BWM_PROBE	IP_HOST	RUNSON	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000
BWM_TX_NODE	IP_HOST	RUNSON	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000
BWM_TX_NODE	BWM_HOST	RUNSON	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000
BWM_TX_NODE	IP_INTERFACE	RUNSON	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000
J2EE_NODE	J2EE_SERVER	PCHILD	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000
J2EE_CELL	J2EE_NODE	PCHILD	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000
J2EE_DOMAIN	J2EE_NODE	PCHILD	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000
J2EE_DOMAIN *	J2EE_SERVER	PCHILD	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000
J2EE_SERVER	IP_HOST	RUNSON	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000
J2EE_SERVER	BWM_HOST	RUNSON	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000

CompTyp_Source_Cd CHAR (17)	CompTyp_Target_Cd CHAR (17)	RelnTyp_Cd CHAR (6)	RelnRul_Strt_DtTm TIMESTAMP	RelnRul_End_DtTm TIMESTAMP
J2EE_SERVER	IP_INTERFACE	RUNSON	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000
IP_HOST	BWM_AGENT_GRP	LCONT	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000
BWM_HOST	BWM_AGENT_GRP	LCONT	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000
IP_INTERFACE	BWM_AGENT_GRP	LCONT	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000
BWM_MGMT_POLICY	BWM_TX_NODE	DESCRI	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000
BWM_MGMT_POLICY	BWM_MGMT_POL_GRP	LCONT	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000

* For WebLogic, which does not have a J2EE_NODE value it has a relationship of J2EE_DOMAIN to J2EE_SERVER that the other J2EE applications cannot have.

7.1.6 Component relationship (table CompReln)

CompReln_ID INTEGER	Comp_Source_ID INTEGER	Comp_Target_ID INTEGER	RelnTyp_Cd CHAR (6)	CompReln_Strt_DtTm TIMESTAMP	CompReln_End_DtTm TIMESTAMP	MSrc_Corr_Cd CHAR (6)
1	3	1	RUNSON	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000	BWM
2	3	4	PCHILD	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000	BWM
3	3	15	INVOKE	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000	BWM
4	8	2	RUNSON	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000	BWM
5	9	1	RUNSON	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000	BWM

CompReIn_ID INTEGER	Comp_Source_ID INTEGER	Comp_Target_ID INTEGER	ReInTyp_Cd CHAR (6)	CompReIn_Strt_DtTm TIMESTAMP	CompReIn_End_DtTm TIMESTAMP	MSrc_Corr_Cd CHAR (6)
6	10	1	RUNSON	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	BWM
7	8	11	USES	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	BWM
8	9	12	USES	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	BWM
9	10	12	USES	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	BWM
10	15	12	USES	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	BWM
11	15	4	INSTOF	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	BWM
12	9	5	INSTOF	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	BWM
13	10	7	INSTOF	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	BWM
14	8	6	INSTOF	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	BWM
15	14	11	PCHILD	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	SHARED
16	13	12	PCHILD	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	SHARED
17	16	14	PCHILD	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	SHARED
18	17	13	PCHILD	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	BWM

CompReIn_ID INTEGER	Comp_Source_ID INTEGER	Comp_Target_ID INTEGER	ReInTyp_Cd CHAR (6)	CompReIn_Strt_DtTm TIMESTAMP	CompReIn_End_DtTm TIMESTAMP	Msrc_Corr_Cd CHAR (6)
				12.00.00.000000	12:00:00.000000	
19	15	1	RUNSON	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	BWM
20	23	9	DESCRI	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	BWM
21	24	10	DESCRI	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	BWM
22	25	8	DESCRI	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	BWM
23	25	27	LCONT	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	BWM
24	23	28	LCONT	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	BWM
25	24	28	LCONT	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	BWM
26	1	26	LCONT	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	BWM
27	2	26	LCONT	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	SHARED
28	18	26	LCONT	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	SHARED
35	11	2	RUNSON	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	SHARED
36	12	1	RUNSON	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	SHARED

7.1.7 Component type keyword (table CompTyp_Keyword)

Keyword_ID INTEGER	CompTyp_Cd CHAR (17)	Keyword_Nm VARCHAR (230)	Keyword_Parent_Nm VARCHAR (230)

7.1.8 Attribute type (table AttrTyp)

AttrTyp_Nm * VARCHAR (120)	AttrTyp_Cd CHAR (17)	MSrc_Corr_Cd CHAR (6)
Last IP Address	LAST_IP_ADDRESS	MODEL1
IP Host Name	IP_HOSTNAME	MODEL1
IP Domain	IP_DOMAIN	MODEL1
IP Network Address	IP_NET_ADDRESS	MODEL1
Node	J2EE_NODE	MODEL1
Web Application Sever	J2EE_SERVER	MODEL1
J2EE Domain	J2EE_DOMAIN	MODEL1
Invoking User	INVOKING_USER	MODEL1
Manufacturer	MANUFACTURER	MODEL1
Version Number	VERSION	MODEL1
Protocol Portion of a URL	URL_PROTOCOL	MODEL1
Website	WEBSITE	MODEL1
Website Path	WEBSITE_PATH	MODEL1

AttrTyp_Nm * VARCHAR (120)	AttrTyp_Cd CHAR (17)	MSrc_Corr_Cd CHAR (6)
Website Query	WEBSITE_QUERY	MODEL1
Warehouse Component Identifier for the Root Transaction	BWM_RT_CDW_ID	BWM
Warehouse Component Identifier for the Parent Transaction	BWM_PT_CDW_ID	BWM
J2EE Cell	J2EE_CELL	BWM
Management Policy	BWM_MGMT_POLICY	BWM
* This column is translated.		

7.1.9 Attribute rule (table AttrRul)

CompTyp_Cd CHAR (17)	AttrTyp_Cd CHAR (17)	AttrRul_Strt_DtTm TIMESTAMP	AttrRul_End_DtTm TIMESTAMP	AttrRul_Dom_Ind CHAR	AttrTyp_Multi_Val CHAR (1)
IP_HOST	LAST_IP_ADDRESS	2002-07-13- 00.00.00..000000	9999-01-01- 12.00.00.000000	N	N
IP_HOST	IP_HOSTNAME	2002-07-13- 00.00.00..000000	9999-01-01- 12.00.00.000000	N	N
BWM_HOST	LAST_IP_ADDRESS	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	N	N
BWM_TX_NODE	URL_PROTOCOL	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	N	N
BWM_TX_NODE	WEBSITE	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	N	N
BWM_TX_NODE	WEBSITE_PATH	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	N	N

CompTyp_Cd CHAR (17)	AttrTyp_Cd CHAR (17)	AttrRul_Strt_DtTm TIMESTAMP	AttrRul_End_DtTm TIMESTAMP	AttrRul_Dom_Ind CHAR	AttrTyp_Multi_Val CHAR (1)
		12.00.00.000000	12:00:00.000000		
BWM_TX_NODE	WEBSITE_QUERY	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	N	N
BWM_TX_NODE	J2EE_NODE	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	N	N
BWM_TX_NODE	J2EE_SERVER	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	N	N
BWM_TX_NODE	INVOKING_USER	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	N	N
BWM_TX_NODE	BWM_MGMT_POLICY	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	N	N
BWM_TX_NODE	J2EE_DOMAIN	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	N	N
BWM_TX_NODE	BWM_RT_CDW_ID	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	N	N
BWM_TX_NODE	BWM_PT_CDW_ID	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	N	Y
J2EE_SERVER	MANUFACTURER	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	N	N
J2EE_SERVER	VERSION	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	N	N

7.1.10 Attribute domain (table AttrDom)

AttrDom_ID INTEGER	CompTyp_Cd CHAR (17)	AttrTyp_Cd CHAR (17)	AttrDom_Strt_DtTm TIMESTAMP	AttrDom_End_DtTm TIMESTAMP	AttrDom_Val VARCHAR (254)	AttrDom_Ds VARCHAR (254)	MSrc_Corr_Cd CHAR (6)
-----------------------	-------------------------	-------------------------	--------------------------------	-------------------------------	---------------------------------	--------------------------------	--------------------------

--	--	--	--	--	--	--	--

7.1.11 Component attribute (table CompAttr)

CompAttr_ID INTEGER	Comp_ID INTEGER	AttrTyp_Cd CHAR (17)	CompAttr_Strt_DfTm TIMESTAMP	CompAttr_End_DfTm TIMESTAMP	CompAttr_Val VARCHAR (254)	Msrc_Corr_Cd CHAR (6)
1	1	LAST_IP_ADDRESS	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000	129.42.16.99	BWM
2	2	LAST_IP_ADDRESS	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000	129.42.18.99	BWM
3	5	URL_PROTOCOL	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000	HTTP	BWM
4	5	WEBSITE	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000	www.ibm.com	BWM
5	5	WEBPATH	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000	hr/index.html	BWM
6	5	WEBQUERY	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000	NULL	BWM
7	7	URL_PROTOCOL	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000	HTTP	BWM
8	7	WEBSITE	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000	www-132.ibm.com:80/	BWM
9	7	WEBPATH	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000	webapp/wcs/stores/servlet/PromotionDisplay	BWM
10	7	WEBQUERY	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000	?promoId=10922&catalogId=-840&storeId=1&langId=-1	BWM
15	8	BWM_MGMT_POLICY	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000	EJBPolicy	BWM
16	8	INVOKING_USER	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000	stewart	BWM
17	8	J2EE_SERVER	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000	Server1	BWM

CompAttr_ID INTEGER	Comp_ID INTEGER	AttrTyp_Cd CHAR (17)	CompAttr_Strt_DtTm TIMESTAMP	CompAttr_End_DtTm TIMESTAMP	CompAttr_Val VARCHAR (254)	Msrc_Corr_Cd CHAR (6)
18	8	J2EE_DOMAIN	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000	jdbc:db2:was40:2003.1.4.15.51.4.539	BWM
19	8	J2EE_NODE	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000	hope	BWM
20	9	BWM_MGMT_POLICY	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000	Policy1	BWM
21	9	INVOKING_USER	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000	db2admin	BWM
22	9	J2EE_SERVER	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000	Server2	BWM
23	9	J2EE_CELL	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000	IBM	BWM
24	9	J2EE_NODE	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000	peace	BWM
25	10	BWM_MGMT_POLICY	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000	Policy9	BWM
26	10	INVOKING_USER	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000	administrator	BWM
27	10	J2EE_SERVER	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000	Server2	BWM
28	10	J2EE_CELL	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000	IBM	BWM
29	10	J2EE_NODE	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000	peace	BWM
30	10	MANUFACTURER	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000	Webshpere	SHARED
31	10	VERSION	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000	5.0	SHARED
32	9	MANUFACTURER	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000	Webshpere	BWM
33	9	VERSION	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000	5.0	BWM
34	8	MANUFACTURER	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000	Webshpere	BWM
35	8	VERSION	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000	4.6	BWM

CompAttr_ID INTEGER	Comp_ID INTEGER	AttrTyp_Cd CHAR (17)	CompAttr_Strt_DfTm TIMESTAMP	CompAttr_End_DfTm TIMESTAMP	CompAttr_Val VARCHAR (254)	MSrc_Corr_Cd CHAR (6)
36	7	BWM_RT_CDW_ID	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000	5	BWM
37	7	BWM_PT_CDW_ID	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000	5	BWM

7.1.12 Component type relationship (table CTypReIn)

CTyp_Source_Cd CHAR (17)	CTyp_Target_Cd CHAR (17)	RelnTyp_Cd CHAR (6)	CompReIn_Strt_DfTm TIMESTAMP	CompReIn_End_DfTm TIMESTAMP

7.1.13 Component attribute type relationship (table ATypReIn)

ATyp_Source_Cd CHAR (17)	ATyp_Target_Cd CHAR (17)	RelnTyp_Cd CHAR (6)	Reln_Strt_DfTm TIMESTAMP	Reln_End_DfTm TIMESTAMP

7.2 Component measurement

The following sections describe the component measurement.

7.2.1 Measurement group type (table MGrpTyp)

MGrpTyp_Cd CHAR (6)	MGrpTyp_Nm * VARCHAR (120)
CATEG	Category
GROUP	Aggregate Types or Group Functions

MGrpTyp_Cd CHAR (6)	MGrpTyp_Nm * VARCHAR (120)
* This column is translated.	

7.2.2 Measurement group (table MGrp)

MGrp_Cd CHAR (6)	MGrpTyp_Cd CHAR (6)	MGrp_Parent_Cd CHAR (6)	MGrp_Nm * VARCHAR (120)
AVG_E	GROUP	NULL	Average Value Exists
MIN_E	GROUP	NULL	Minimum Value Exists
MAX_E	GROUP	NULL	Maximum Value Exists
TOT_E	GROUP	NULL	Total Value Exists
* This column is translated.			

7.2.3 Measurement group member (table MGrpMbr)

MGrp_Cd CHAR (6)	MGrpTyp_Cd CHAR (6)	MsmtTyp_ID INTEGER
AVG_E	GROUP	1
AVG_E	GROUP	2
AVG_E	GROUP	3
AVG_E	GROUP	4
AVG_E	GROUP	6
AVG_E	GROUP	7

MGrp_Cd CHAR (6)	MGrpTyp_Cd CHAR (6)	MsmfTyp_ID INTEGER
MAX_E	GROUP	1
MAX_E	GROUP	2
MAX_E	GROUP	3
MAX_E	GROUP	4
MAX_E	GROUP	6
MAX_E	GROUP	7
MIN_E	GROUP	1
MIN_E	GROUP	2
MIN_E	GROUP	3
MIN_E	GROUP	4
MIN_E	GROUP	6
MIN_E	GROUP	7
TOT_E	GROUP	5
TOT_E	GROUP	8
TOT_E	GROUP	9

7.2.4 Measurement unit category (table MUnitCat)

This warehouse pack does not use the measurement unit category table.

7.2.5 Measurement unit (table MUnit)

MUnit_Cd CHAR (6)	MUnitCat_Cd CHAR (6)	Munit_Nm * VARCHAR (120)
PRC	PRC	Percentage
QTY	QTY	Quantity
Sec	TM	Seconds
MSec	TM	Milliseconds
* This column is translated.		

7.2.6 Measurement alias names (table MTypReIn)

MTyp_Source_ID INTEGER	MTyp_Target_ID INTEGER	RelnTyp_Cd CHAR (6)	CompReIn_Strt_DtTm TIMESTAMP	CompReIn_End_DtTm TIMESTAMP	

7.2.7 Time summary (table TmSum)

The period over which a measurement may be summarized.

TmSum_Cd CHAR	TmSum_Nm * VARCHAR (120)
H	Hourly
* This column is translated.	

7.2.8 Measurement source (table MSrc)

MSrc_Cd CHAR (6)	MSrc_Parent_Cd CHAR (6)	MSrc_Nm VARCHAR (120)

MSrc_Cd CHAR (6)	MSrc_Parent_Cd CHAR (6)	MSrc_Nm VARCHAR (120)
Tivoli	NULL	Tivoli Application
BWM	Tivoli	IBM Tivoli Monitoring for Transaction Performance v 5.3.0

7.2.9 Measurement source history (table MSrcHistory)

MSrc_Cd CHAR (6)	MSrc_Nm VARCHAR (120)	MSrc_Strt_DtTm TIMESTAMP	MSrc_End_DtTm TIMESTAMP

7.2.10 Measurement type (table MsmtTyp)

MsmtTyp_ID INTEGER	MUnit_Cd CHAR (6)	MSrc_Cd CHAR (6)	MsmtTyp_Nm * VARCHAR (120)	MsmtTyp_Ds * VARCHAR (254)
1	MSec	MODEL1	Response Time**	The amount of time it took a process to respond
2	Sec	BWM	Round Trip Time	Round trip transaction response time
3	Sec	BWM	Service Time	Backend service transaction response time
4	Sec	BWM	Page Render Time	Page render transaction response time
5	QTY	BWM	Number Threshold Exceeded	Number of thresholds exceeded
6	PRC	BWM	Successful Transactions	Percentage of successful synthetic transactions
7	PRC	BWM	Unsuccessful Transactions	Percentage of unsuccessful synthetic transactions

MsmtTyp_ID INTEGER	MUnit_Cd CHAR (6)	MSrc_Cd CHAR (6)	MsmtTyp_Nm * VARCHAR (120)	MsmtTyp_Ds * VARCHAR (254)
8	QTY	BWM	Number of Executions	Number of times a transaction was executed
9	QTY	BWM	Number of Aborted Transactions	Number of times a transaction was aborted
* This column is translated.				

The measurements for 'Number of Executions', 'Number of Aborted Transactions', 'Number Threshold Exceeded' are totals of all the counts for the period. The other measurements are averages of the values for the period.

7.2.11 Component measurement rule (table MsmtRul)

CompTyp_Cd CHAR (17)	MsmtTyp_ID INTEGER
BWM_TX_NODE	1
BWM_TX_NODE	2
BWM_TX_NODE	3
BWM_TX_NODE	4
BWM_TX_NODE	5
BWM_TX_NODE	6
BWM_TX_NODE	7
BWM_TX_NODE	8
BWM_TX_NODE	9

7.2.12 Measurement (table Msmt)

Msmt_ID BIGINT	Comp_ID INTEGER	MsmtTyp_ID INTEGER	TmSum_Cd CHAR	Msmt_Strt_Dt DATE	Msmt_Strt_Tm TIME	Msmt_Min_Val FLOAT	Msmt_Max_Val FLOAT	Msmt_Avg_Val FLOAT	Msmt_Tot_Val FLOAT	Msmt_Smpl_Cnt INTEGER	Msmt_Err_Cnt INTEGER	msmt_stddev_Val DOUBLE	MSrc_Corr_Cd CHAR (6)
1	9	1	H	2002-06-30	13:00:00	300	1078	502		248	8		BWM
2	9	2	H	2002-06-30	13:00:00	358	6052	3054		248	8		BWM
3	9	5	H	2002-06-30	13:00:00				20	256	0		BWM
4	8	1	H	2002-06-30	13:00:00	300	1078	502		253	3		BWM
5	8	2	H	2002-06-30	13:00:00	358	6052	3054		253	3		BWM
6	8	5	H	2002-06-30	13:00:00				15	255	1		BWM
7	15	1	H	2002-06-30	13:00:00	100	200	150		356	0		BWM
8	15	2	H	2002-06-30	13:00:00	406	1000	675		356	0		BWM

Msmt_ID BIGINT	Comp_ID INTEGER	MsmtTyp_ID INTEGER	TmSum_Cd CHAR	Msmt_Strt_Dt DATE	Msmt_Strt_Tm TIME	Msmt_Min_Val FLOAT	Msmt_Max_Val FLOAT	Msmt_Avg_Val FLOAT	Msmt_Tot_Val FLOAT	Msmt_Smpl_Cnt INTEGER	Msmt_Err_Cnt INTEGER	mmt_stddev_Val DOUBLE	MSrc_Corr_Cd CHAR (6)
9	15	3	H	2002-06-30	13:00:00	26	6300	5000		356	0		BWM
10	10	1	H	2002-06-30	14:00:00	100	200	150		356	0		BWM
11	10	2	H	2002-06-30	14:00:00	406	1000	675		356	0		BWM
12	10	3	H	2002-06-30	14:00:00	26	6300	5000		356	0		BWM

7.2.13 Threshold measurement objective (table Mobj)

Mobj_ID INTEGER	MsmtTyp_ID INTEGER	CompTyp_Cd CHAR (17)	Centr_Cd CHAR (6)	Cust_ID INTEGER	Attrdom_ID INTEGER	MSrc_CHAR (6)	Mobj_Strt_DtTm TIMESTAMP	Mobj_End_DtTm TIMESTAMP

7.2.14 Threshold measurement objective range (table MobjRng)

Mobjrng_ID INTEGER	Mobj_ID INTEGER	Sev_Cd CHAR	Mobjrng_Min_Val FLOAT	Mobjrng_Max_Val DOUBLE	Mobjrng_Strt_Dow TIMESTAMP	Mobjrng_End_Dow TIMESTAMP	Mobjrng_Strt_Tm TIMESTAMP	Mobjrng_End_Tm TIMESTAMP

7.2.15 Threshold severity level (table SevLvl)

Sev_Cd CHAR	MSrc_CHAR (6)	Sev_Nm * VARCHAR (254)
* This column is translated.		

7.3 Helper tables

The following are helper tables for IBM Tivoli Monitoring for Transaction Performance.

Note: The data samples provided in the following tables do not match the other data samples provided in this document.

7.3.1 Component long (table BWM.COMP_ATTR_LONG)

The component long table is used to store component attributes value that are longer than the 254 characters allowed in the component attribute table (CompAttr).

Compattr_ID INTEGER	Comp_ID INTEGER	Compattr_val VARCHAR (4000)	AttrTyp_CD VARCHAR(17)

73	1972	?forwardName=MQueue.content.main&sfname=factories&resourceUri=resources.xml&parentRefId=builtin_mqprovider&contextId=cells:stewart:nodes:stewart&perspective=tab.configuration	WEBSITE_QUERY
----	------	--	---------------

7.4 Exception tables

One exception table, BWM.Exception_Log, is provided. It is used to identify cases where data might fall outside of an allowable, expected range of values or cases where extract run-time problems were encountered and logged. The data written to the table identifies the table, columns and the calculation performed in which the exception was found, the range of values expected, as well as the actual values found.

Error_DtTm TIMESTAM P	Process_Nm VARCHAR(120)	Step_Nm VARCHAR(120)	Table_Nm VARCHAR(120)	Calculation VARCHAR(120)	Result_Val_Low VARCHAR(120)	Result_Val_High VARCHAR(254)	Actual_Value VARCHAR(254)	Error_Msg_Text VARCHAR(254)	Error_Recovery VARCHAR(254)	Reported_DtTm TIMESTAMP

7.5 Incremental extraction

The extraction processes typically run once in a 24-hour period. Each ETL process extracts only the data that has been added to the database since the previous successful completion of the ETL processes. Incremental extraction prevents excessive use of time and resources, which would result if all the data were re-extracted for each extraction.

This warehouse pack uses incremental extraction to do the following:

- Extract data from the Tivoli Monitoring for Transaction Performance source database and store it into the central data warehouse during central data warehouse ETL (ETL1)
- Extract data from the central data warehouse and store it into the data mart tables during data mart ETL (ETL2)

In table TWG.Extract_Control columns EXTCTL_FROM_INTSEQ and EXTCTL_TO_INTSEQ control the range of data pulled during the prior extraction process. After the ETL process completes, the EXTCTL_FROM_INTSEQ is reset to reference where the extraction should begin the next time the ETL processes are run. These numbers should be equal.

This warehouse pack uses the TWG.Extract_Control table to hold the extraction values for both the central data warehouse ETL (ETL1) and the data mart ETL (ETL2) extractions. The following table shows the columns used in the TWG.Extract_Control table and the values this table has when this warehouse pack is first installed and the central data warehouse ETL (ETL1) process has not yet been run. The -1 values in the EXTCTL_FROM_INTSEQ and EXTCTL_TO_INTSEQ columns indicate that all data from the source table should be extracted. The source name, AGGREGATEDATA, listed in EXTCTL_SOURCE is the table extracted from during the central data warehouse ETL (ETL1) process. The rest are the table names are the source tables extracted from during the data mart ETL (ETL2) process. See section 5.3, Extraction control (tableExtract_Control), for more details on this process.

EXTCTL_SOURCE	EXTCTL_TARGET	EXTCTL_FROM_INTSEQ	EXTCTL_TO_INTSEQ
1.AGGREGATEDATA	BWM.STAGE AGGREG DATA	-1	-1
2.AGGREGATEDATA	BWM.STAGE AGGREG DATA	-1	-1
TWG.MSMT	BWM.STG1 TX ND HR	-1	-1
TWG.MSMT	BWM.STG2 TX ND HR	-1	-1
BWM.STG TX ND MET	BWM.T TX ND METRIC	-1	-1
TWG.COMP	BWM.T_APP	-1	-1
TWG.COMP	BWM.T_HOST	-1	-1
TWG.COMP	BWM.T_TX	-1	-1
TWG.COMP	BWM.T_TX_ND	-1	-1
BWM.COMP_NAME_LONG	BWM.COMP_NAME_LONG	-1	-1

BWM.COMP_ATTR_LONG	BWM.COMP_ATTR_LONG	-1	-1
--------------------	--------------------	----	----

8 Data mart schema information

The following sections contain the definition of star schemas, metric dimension tables, and data marts provided with the warehouse pack. This section is intended primarily for report designers and warehouse pack creators. For information about reports, see “Reports” on page 21.

Shaded columns in the following tables are translated. These columns are also marked with an asterisk (*) after the column name.

8.1 Data mart BWM Transaction Performance

This data mart uses the following star schemas:

- BWM_Hourly_Transaction_Node_Star_Schema
- BWM_Daily_Transaction_Node_Star_Schema
- BWM_Weekly_Transaction_Node_Star_Schema
- BWM_Monthly_Transaction_Node_Star_Schema

8.2 Star schemas

Before using this section, read about the star schemas in *Enabling an Application for Tivoli Data Warehouse*. That document defines the content of each table and explains the relationships between the tables in this document.

The warehouse pack provides the following star schemas.

8.2.1 BWM hourly transaction performance transaction node star schema

The following table defines the star schema. The description of the star schema is translated.

Description of star schema (in IWH_STARSHEMA)	BWM hourly transaction performance transaction node star schema
Name of fact table	BWM.F_TX_ND_HOUR
Name of metric dimension table	BWM.D_TX_ND_METRIC
Names of other dimension tables	BWM.D_HOST

Description of star schema (in IWH_STARSHEMA)	BWM hourly transaction performance transaction node star schema
	BWM.D_HOST_AGENT_GRPS
	BWM.D_TX_ND_POL_GRPS
	BWM.D_TX
	BWM.D_TX_ND
	BWM.D_APP

8.2.1.1 Fact table BWM.F_TX_ND_HOUR

The following columns are used in the fact table:

- Fact_ID INTEGER
- CDW_ID INTEGER
- Metric_ID INTEGER
- Host_ID INTEGER
- TX_ID INTEGER
- TX_ND_ID INTEGER
- App_ID INTEGER
- Meas_hour TIMESTAMP
- Min_value DOUBLE
- Max_value DOUBLE
- Avg_value DOUBLE
- Total_value DOUBLE
- Sample_count BIGINT
- Error_count BIGINT

8.2.2 BWM daily transaction performance transaction node star schema

The following table defines the star schema. The description of the star schema is translated.

Description of star schema (in IWH_STARSHEMA)	BWM daily transaction performance transaction node star schema
Name of fact table	BWM.F_TX_ND_DAY
Name of metric dimension table	BWM.D_TX_ND_METRIC
Names of other dimension tables	BWM.D_HOST
	BWM.D_HOST_AGENT_GRPS
	BWM.D_TX_ND_POL_GRPS
	BWM.D_TX
	BWM.D_TX_ND
	BWM.D_APP

8.2.2.1 Fact table BWM.F_TX_ND_DAY

- Fact_ID INTEGER
- CDW_ID INTEGER
- Metric_ID INTEGER
- Host_ID INTEGER
- TX_ID INTEGER
- TX_ND_ID INTEGER
- App_ID INTEGER
- Meas_date TIMESTAMP

- Min_value DOUBLE
- Max_value DOUBLE
- Avg_value DOUBLE
- Total_value DOUBLE
- Sample_count BIGINT
- Error_count BIGINT

8.2.3 BWM weekly transaction performance transaction node star schema

The following table defines the star schema. The description of the star schema is translated.

Description of star schema (in IWH_STARSHEMA)	BWM weekly transaction performance transaction node star schema
Name of fact table	BWM.F_TX_ND_WEEK
Name of metric dimension table	BWM.D_TX_ND_METRIC
Names of other dimension tables	BWM.D_HOST
	BWM.D_HOST_AGENT_GRPS
	BWM.D_TX_ND_POL_GRPS
	BWM.D_TX
	BWM.D_TX_ND
	BWM.D_APP

8.2.3.1 Fact table BWM.F_TX_ND_WEEK

- Fact_ID INTEGER
- CDW_ID INTEGER

- Metric_ID INTEGER
- Host_ID INTEGER
- TX_ID INTEGER
- TX_ND_ID INTEGER
- App_ID INTEGER
- Meas_date TIMESTAMP
- Min_value DOUBLE
- Max_value DOUBLE
- Avg_value DOUBLE
- Total_value DOUBLE
- Sample_count BIGINT
- Error_count BIGINT

8.2.4 BWM monthly transaction performance transaction node star schema

The following table defines the star schema. The description of the star schema is translated.

Description of star schema (in IWH_STARSHEMA)	BWM monthly transaction performance transaction node star schema
Name of fact table	BWM.F_TX_ND_MONTH
Name of metric dimension table	BWM.D_TX_ND_METRIC
Names of other dimension tables	BWM.D_HOST
	BWM.D_HOST_AGENT_GRPS
	BWM.D_TX_ND_POL_GRPS
	BWM.D_TX

Description of star schema (in IWH_STARSHEMA)	BWM monthly transaction performance transaction node star schema
	BWM.D_TX_ND
	BWM.D_APP

8.2.4.1 Fact table BWM.F_TX_ND_MONTH

- Fact_ID INTEGER
- CDW_ID INTEGER
- Metric_ID INTEGER
- Host_ID INTEGER
- TX_ID INTEGER
- TX_ND_ID INTEGER
- App_ID INTEGER
- Meas_date TIMESTAMP
- Min_value DOUBLE
- Max_value DOUBLE
- Avg_value DOUBLE
- Total_value DOUBLE
- Sample_count BIGINT
- Error_count BIGINT

8.2.5 AMY IBM Tivoli Monitoring for Operating Systems star schema for cross application reporting

The following table defines the star schema. The description of the star schema is translated.

8.1.1 AMY Hourly/Daily/Weekly/Monthly CPU Star Schema

Description of star schema (in IWH.STARSCHEMA)	Star schema for all the CPU related metrics
Name of fact table	AMY.F_CPU_HOUR, AMY.F_CPU_DAY, AMY.F_CPU_WEEK, AMY.F_CPU_MONTH
Name of metric dimension table	AMY.D_CPU_METRIC
Names of other dimension tables	AMY.D_HOST_IP, AMY.D_CPU

Fact table AMY.F_CPU_HOUR

Metric_ID INTEGER	Host_ID INTEGER	HOST_IP_AD_ID INTEGER	CPU_ID INTEGER	Meas_hour TIME STAMP	Min_value DOUBLE	Max_value DOUBLE	Avg_value DOUBLE	Total_value DOUBLE	Sample_count DOUBLE
0	0	0	0	2002-08-29 04:00:17	0.00	3.00	0.38	NULL	NULL

Fact table AMY.F_MEMORY_HOUR

Metric_ID INTEGER	Host_ID INTEGER	HOST_IP_AD_ID INTEGER	MEMORY_ID INTEGER	Meas_hour TIME STAMP	Min_value DOUBLE	Max_value DOUBLE	Avg_value DOUBLE	Total_value DOUBLE	Sample_count DOUBLE
0	0	0	0	2002-08-29 04:00:17	0	1,54E-1	3,17E-3	NULL	NULL

8.3 Fact staging tables

The following section describes the fact staging table used by this warehouse pack.

8.3.1 Fact staging table BWM.STG_TX_ND_HR

The fact staging table supports the conversion of data from multiple central data warehouses into one table. It contains the data extracted from the central data warehouses before it is loaded into the hourly fact table. Each hourly fact table has a corresponding fact staging table. The fact staging table for this warehouse pack is BWM.STG_TX_ND_HR and has the following layout:

- CDW_ID INTEGER
- Msmt_ID BIGINT
- ORIG_TX_ND_ID INTEGER
- ORIG_TX_ID INTEGER
- ORIG_App_ID INTEGER
- ORIG_Host_ID INTEGER
- ORIG_Metric_ID INTEGER
- Meas_hour TIMESTAMP
- Min_value DOUBLE
- Max_value DOUBLE
- Avg_value DOUBLE
- Total_value DOUBLE
- Sample_count BIGINT
- Error_count BIGINT

PRIMARY KEY (CDW_ID, MSMT_ID)

8.4 Metric dimension tables

This section describes the metric dimension tables used by the star schemas in the warehouse pack. Shaded columns indicate text that is translated. These column headings are also marked with an asterisk (*).

8.4.1 BWM.D_TX_ND_METRIC

The table below shows the layout of the dimension metric table for this warehouse pack. It defines the metrics available and which types of measurements are available for each metric. The following table is the metric table for this warehouse pack:

- BWM.T_TX_ND_METRIC

Metric_ID INTEGER	Met_Category * VARCHAR (254)	Met_Desc * VARCHAR (254)	Met_Name * VARCHAR (254)	Met_Units * VARCHAR (254)	Min_Exists CHAR (1)	Max_Exists CHAR (1)	Avg_Exists CHAR (1)	Total_Exists CHAR (1)	Msrc_Nm * VARCHAR (254)
1	Not Used	Number of Times a transaction was executed	Number of Executions	QTY	N	N	N	Y	IBM Tivoli Monitoring For Transaction Performance v 5.3
2	Not Used	The backend service response time	Service Time	Sec	Y	Y	Y	N	IBM Tivoli Monitoring For Transaction Performance v 5.3
3	Not Used	The number of transaction thresholds exceeded	Number Threshold Exceeded	QTY	N	N	N	Y	IBM Tivoli Monitoring For Transaction Performance v 5.3
4	Not Used	The page render response time	Page Render Time	Sec	Y	Y	Y	N	IBM Tivoli Monitoring For Transaction Performance v 5.3
5	Not Used	The percentage of synthetic transactions that failed	Unsuccessful Transactions	PRC	Y	Y	Y	N	IBM Tivoli Monitoring For Transaction Performance v 5.3
6	Not Used	The percentage of synthetic transactions that were successful	Successful Transactions	PRC	Y	Y	Y	N	IBM Tivoli Monitoring For Transaction Performance v 5.3
7	Not Used	The amount of time it took a process to respond	Response Time	MSec	Y	Y	Y	N	IBM Tivoli Common Data Model v 1

Metric_ID INTEGER	Met_Category * VARCHAR (254)	Met_Desc * VARCHAR (254)	Met_Name * VARCHAR (254)	Met_Units * VARCHAR (254)	Min_Exists CHAR (1)	Max_Exists CHAR (1)	Avg_Exists CHAR (1)	Total_Exists CHAR (1)	Msrc_Nm * VARCHAR (254)
8	Not Used	Number of Times a transaction was aborted	Number of Aborted Transactions	QTY	N	N	N	Y	IBM Tivoli Monitoring For Transaction Performance v 5.3
9	Not Used	The round trip response time	Round Trip Time	Sec	Y	Y	Y	N	IBM Tivoli Monitoring For Transaction Performance v 5.3
* This column is translated.									

8.4.2 AMY.D_CPU_METRIC

The table below shows the layout of the AMY dimension metric table used by this warehouse pack for the cross application report. It defines the metric used by this warehouse for the cross application report to obtain CPU usage metrics.

- AMY.D_CPU_METRIC
- AMY.D_MEM_METRIC

AMY.D_CPU_METRIC

This table stores only the CPU related metrics, i.e. the metrics belonging to the CPU_E measurement group defined into the IBM Tivoli Monitoring Collection measurement group

Metric_ID INTEGER	Met_Category * VARCHAR (254)	Met_Desc * VARCHAR (254)	Met_Name * VARCHAR (254)	Met_Units * VARCHAR (254)	Min_Exists CHAR (1)	Max_Exists CHAR (1)	Avg_Exists CHAR (1)	Total_Exists CHAR (1)	Msrc_Nm * VARCHAR (254)
1	CPU	Percentage of overall CPU utilization	basicAverageCPUPct	PRC	Y	Y	Y	N	IBM Tivoli Monitoring For Operating Systems

AMY.D_MEM_METRIC

This tables stores only the memory related metrics, i.e. the metrics belonging to the MEM_E measurement group defined into the IBM Tivoli Monitoring Collection measurement group

Metric_ID INTEGER	Met_Category * VARCHAR (254)	Met_Desc * VARCHAR (254)	Met_Name * VARCHAR (254)	Met_Units * VARCHAR (254)	Min_Exists CHAR (1)	Max_Exists CHAR (1)	Avg_Exists CHAR (1)	Total_Exists CHAR (1)	Msrc_Nm * VARCHAR (254)
1	Memory	Amount of committed bytes in relation to the total physical memory (RAM) available to the operating system.	PercentMemoryUsage	PRC	Y	Y	Y	N	IBM Tivoli Monitoring For Operating Systems

8.5 Dimension tables

The following sections describe the dimension tables (other than metric dimension tables) used by the star schemas in the warehouse pack.

8.5.1 Dimension table BWM.D_HOST

The following columns are used in this dimension table.

- Host_ID INTEGER
- Host_NM VARCHAR
- IP_Address VARCHAR
- Cust_ID INTEGER
- Cust_NM VARCHAR
- Center_NM VARCHAR

8.5.2 Dimension table BWM.D_HOST_AGENT_GRP

- Host_ID INTEGER
- Agent_Group VARCHAR

8.5.3 Dimension table BWM.D_TX

The following columns are used in this dimension table.

- TX_ID INTEGER
- TX_Name VARCHAR
- TX_Descr VARCHAR
- URL_Protocol VARCHAR
- Website VARCHAR
- WebSite_Path VARCHAR
- Website_Query VARCHAR

8.5.4 Dimension table BWM.D_TX_ND

The following columns are used in this dimension table.

- TX_ND_ID INTEGER
- TX_ND_Name VARCHAR

- CURRENT_CDW_ID VARCHAR
- ROOT_CDW_ID VARCHAR
- PARENT_CDW_ID VARCHAR
- Management_Policy VARCHAR
- Invoking_User VARCHAR

8.5.5 Dimension table BWM.D_TX_ND_POL_GRP

- TX_ND_ID INTEGER
- Policy_Group VARCHAR

8.5.6 Dimension table BWM.D_APP

The following columns are used in this dimension table.

- App_ID INTEGER
- J2EE_Server_Name VARCHAR
- J2EE_Cell_Name VARCHAR
- J2EE_Domain VARCHAR
- J2EE_Node VARCHAR
- J2EE_Type_Ver VARCHAR
- PROBE_NAME VARCHAR
- PROBE_HOST VARCHAR

8.5.7 AMY.D_HOST_IP

This table stores the same information of the previous table, adding fields for Customer Name and Center Name and removing the information about the life period of the host, because it is always filled with data already filtered by the TWG.Cur_Comp view, which only consider the active components.

Host_ID INTEGER	Hostname VARCHAR(120)	Network_domain VARCHAR(120)	Network_subdomain VARCHAR(120)	Network_subdomain2 VARCHAR(120)	Short_hostname VARCHAR(120)	Customer_Name VARCHAR(120)	Center_Name VARCHAR(120)
0	dmw2k3.rome.tivoli.com	tivoli.com	rome.tivoli.com	no value	dmw2k3	Default CDW customer	Default CDW center

8.6 Data translation tables

The following tables are utilized by the data mart ETL to move data from the central data warehouse database to the data mart database using extract control to only extract the newly added data since the last data mart ETL run. Each dimension table has a corresponding translation table. The translation tables for this warehouse pack are:

- BWM.T_APP
- BWM.T_HOST
- BWM.T_TX
- BWM.T_TX_ND

8.6.1 BWM.T_TX_ND_METRIC

The following columns are used in this translation table:

- orig_metric_id INTEGER
- metric_id INTEGER
- cdw_id INTEGER
- met_category VARCHAR
- met_desc VARCHAR

- met_name VARCHAR
- met_units VARCHAR
- min_exists CHAR
- max_exists CHAR
- avg_exists CHAR
- total_exists CHAR
- msrc_nm VARCHAR

8.6.2 BWM.T_HOST

The following columns are used in this translation table:

- Orig_Host_ID INTEGER
- CDW_ID INTEGER
- Host_NM VARCHAR
- IP_Address VARCHAR
- Cust_ID INTEGER
- Cust_NM VARCHAR
- Center_NM VARCHAR

8.6.3 BWM.T_HOST_AGENT_GRP

- Host_ID INTEGER
- Agent_Group VARCHAR

8.6.4 BWM.T_TX

The following columns are used in this translation table:

- Orig_TX_ID INTEGER
- CDW_ID INTEGER
- TX_Name VARCHAR

- TX_Description VARCHAR
- URL_Protocol VARCHAR
- Website VARCHAR
- WebSite_Path VARCHAR
- Website_Query VARCHAR

8.6.5 BWM.T_TX_ND

The following columns are used in this translation table:

- Orig_TX_ND_ID INTEGER
- CDW_ID INTEGER
- TX_ND_Name VARCHAR
- ROOT_CDW_ID VARCHAR
- PARENT_CDW_ID VARCHAR
- Management_Policy VARCHAR
- Invoking_User VARCHAR

8.6.6 BWM.T_TX_ND_POL_GRP

The following columns are used in this translation table:

- TX_ND_ID INTEGER
- Policy_Group VARCHAR

8.6.7 BWM.T_APP

The following columns are used in this translation table:

- OrigApp_id INTEGER
- CDW_ID INTEGER
- J2EEServer_Name VARCHAR
- J2EE_Cell_Name VARCHAR

- J2EE_Domain VARCHAR
- J2EE_Node VARCHAR
- J2EE_Type_Ver VARCHAR
- PROBE_NAME VARCHAR
- PROBE_HOST VARCHAR

8.7 Exception tables

One exception table, BWM.Exception_Log, is provided. It is used to identify cases where data might fall outside of an allowable, expected range of values or cases where extract run-time problems were encountered and logged. The data written to the table identifies the table, columns and the calculation performed in which the exception was found, the range of values expected, as well as the actual values found.

Error_DtTm TIMESTAM P	Process_Nm VARCHAR(120)	Step_Nm VARCHAR(120)	Table_Nm VARCHAR(120)	Calculation VARCHAR(120)	Result_Val_Low VARCHAR(120)	Result_Val_High VARCHAR(254)	Actual_Value VARCHAR(254)	Error_Msg_Text VARCHAR(254)	Error_Recovery VARCHAR(254)	Reported_DtTm TIMESTAMP

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 grant 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 may 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/or 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
224A/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.

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 illustrates 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 softcopy, the photographs and color illustrations may not appear.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM, the IBM logo, Tivoli, the Tivoli logo, AIX, DB2, DRDA, Informix, OS/2, OS/400, Tivoli Enterprise Console, and TME are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

Intel, the Intel Inside logos, MMX, and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

Microsoft and Windows are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

SET and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.



Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.



Printed in U.S.A.

SC32-9109-00