



**IBM Tivoli Monitoring for Transaction  
Performance, Version 5.2  
Warehouse Enablement Pack, Version 5.2.0  
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 96.

**Second Edition (June 2004)**

This edition applies to IBM Tivoli Monitoring for Transaction Performance, Version 5.2 and to all subsequent releases and modifications until otherwise indicated in new editions.

© Copyright International Business Machines Corporation 2003, 2004. All rights reserved.

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

# Contents

<b>1 About this guide .....</b>	<b>7</b>
1.1 What's new in this warehouse pack .....	7
1.2 Who should read this guide .....	7
1.3 Publications .....	8
1.3.1 IBM Tivoli Monitoring for Transaction Performance library.....	8
1.3.2 Tivoli Data Warehouse library .....	9
1.3.3 Related publications.....	9
1.3.3.1 IBM Redbooks.....	9
1.3.3.2 IBM DB2, DB2 Data Warehouse Center, and DB2 Warehouse Manager library .....	10
1.3.4 Accessing publications online .....	10
1.3.5 Ordering publications .....	11
1.4 Accessibility.....	11
1.5 Contacting software support .....	11
1.6 Participating in newsgroups .....	11
1.7 Typeface conventions .....	11
 <b>2 Overview .....</b>	 <b>13</b>
2.1 Overview of Tivoli Data Warehouse .....	13
2.2 Overview of IBM Tivoli Monitoring for Transaction Performance Version 5.2 warehouse pack .....	14
 <b>3 Reports.....</b>	 <b>18</b>
3.1 Report Parameter Inputs Defined.....	31
3.1.1 Average Response Time by Application .....	32
3.1.2 Average Response Time by User.....	32
3.1.3 Average Response Time by Policy .....	32
3.1.4 Average Response Time by Agent .....	33
3.1.5 Execution Load by Application .....	34
3.1.6 Execution Load by User .....	34
3.1.7 Execution Load For Policy <policy>.....	35
3.1.8 Execution Load per Agent For Policy <policy> .....	33
3.1.9 Transaction Availability For Transaction <transaction> .....	35
3.1.10 Overall Transaction Response For Transaction <transaction> .....	36
3.1.11 Policy Load per Agent.....	36
3.1.12 Slowest Throughput.....	37
3.1.13 Slowest Transactions .....	37
 <b>4 Installing and configuring the warehouse pack .....</b>	 <b>40</b>
4.1 Prerequisite hardware and software.....	40
4.1.1 Installing the Crystal Enterprise 9 patch .....	40
4.2 Product notes and limitations.....	41
4.2.1 Product notes .....	41
4.2.2 Limitations.....	41
4.2.2.1 No support for DB2 Universal Database for z/OS and OS/390.....	41
4.2.2.2 Reinstalling the IBM Tivoli Monitoring for Transaction Performance warehouse pack.....	41
4.2.2.3 Initial Data Extraction.....	41
4.2.2.4 How to handle multiple failed unattended executions of the ETL.....	42
4.3 Database-sizing considerations.....	43
4.4 Pre-installation procedures.....	43
4.4.1 Tivoli Data Warehouse, Version 1.2 requirement .....	43
4.4.2 Creating warehouse database backups.....	43

4.4.3 Upgrading to Version 5.2.0 of the IBM Tivoli Monitoring for Transaction Performance warehouse pack ..	44
4.4.4 ODBC database client.....	44
4.4.5 ODBC drivers .....	44
<b>4.5 Installation of the warehouse pack.....</b>	<b>44</b>
<b>4.6 Post-installation procedures .....</b>	<b>45</b>
4.6.1 Scheduling warehouse pack processes.....	45
4.6.2 Change Prune Control values from installed defaults.....	46
4.6.3 Accessing the application source databases.....	46
4.6.4 Configuring warehouse source and target settings .....	46
<b>4.7 Migration from a previous release of the warehouse pack .....</b>	<b>47</b>
<b>4.8 Un-installation of the warehouse pack.....</b>	<b>47</b>
<b>4.9 Multiple data centers.....</b>	<b>49</b>
<b>4.10 Multiple customer environments.....</b>	<b>49</b>
 <b>5 Maintenance and problem determination .....</b>	 <b>50</b>
<b>5.1 Backing up and restoring.....</b>	<b>50</b>
<b>5.2 Pruning data.....</b>	<b>50</b>
5.2.1 Central data warehouse .....	50
5.2.1.1 Pruning measurement data (table Prune_Msmt_Control) .....	50
5.2.2 Data mart .....	51
5.2.2.1 Pruning mart data (table Prune_Mart_Control) .....	51
<b>5.3 Extraction control (table Extract_Control).....</b>	<b>52</b>
5.3.1 Msmt data .....	52
5.3.2 Data mart data.....	53
<b>5.4 Maintenance scripts.....</b>	<b>53</b>
5.4.1 Show_ExtCtl_Values.sql .....	53
5.4.2 Show_ExtLog_Values.sql .....	54
5.4.3 Show_Prune_Mart_Values.sql .....	54
5.4.4 Show_Prune_Msmt_Values.sql.....	54
5.4.5 Reset_ETL1_extract_window.sql.....	54
5.4.6 Reset_ETL2_extract_window.sql.....	54
5.4.7 Reset_ETL1_ETL2_extract_windows.sql .....	55
5.4.8 Reset_ETL1_ETL2_prune_values_to_clear_data.sql.....	55
5.4.9 Reset_ETL1_ETL2_prune_values_to_defaults.sql .....	56
<b>5.5 Problem determination .....</b>	<b>56</b>
 <b>6 ETL processes.....</b>	 <b>57</b>
<b>6.1 BWM_c10_CDW_Process .....</b>	<b>57</b>
<b>6.2 BWM_m05_Mart_Process.....</b>	<b>57</b>
 <b>7 Central data warehouse information .....</b>	 <b>59</b>
<b>7.1 Component configuration .....</b>	<b>59</b>
7.1.1 Component type (table CompTyp) .....	59
7.1.2 Component extension (table Comp_ext) .....	60
7.1.3 Component (table Comp) .....	60
7.1.4 Component relationship type (table RelnTyp).....	63
7.1.5 Component relationship rule (table RelnRul) .....	64
7.1.6 Component relationship (table CompReln) .....	65
7.1.7 Component type keyword (table CompTyp_Keyword).....	67
7.1.8 Attribute type (table AttrTyp).....	67
7.1.9 Attribute rule (table AttrRul) .....	68
7.1.10 Attribute domain (table AttrDom) .....	70
7.1.11 Component attribute (table CompAttr).....	70
7.1.12 Component type relationship (table CTypReln) .....	73

7.1.13 Component attribute type relationship (table ATypReln).....	73
<b>7.2 Component measurement .....</b>	<b>73</b>
7.2.1 Measurement group type (table MGrpTyp).....	73
7.2.2 Measurement group (table MGrp) .....	74
7.2.3 Measurement group member (table MGrpMbr) .....	74
7.2.4 Measurement unit category (table MUnitCat) .....	75
7.2.5 Measurement unit (table MUnit) .....	76
7.2.6 Measurement alias names (table MTypReln) .....	76
7.2.7 Time summary (table TmSum).....	76
7.2.8 Measurement source (table MSrc).....	77
7.2.9 Measurement source history (table MSrcHistory) .....	77
7.2.10 Measurement type (table MsmtTyp).....	77
7.2.11 Component measurement rule (table MsmtRul) .....	78
7.2.12 Measurement (table Msmt).....	79
7.2.13 Threshold measurement objective (table Mobj) .....	80
7.2.14 Threshold measurement objective range (table MobjRng).....	81
7.2.15 Threshold severity level (table SevLvl).....	81
<b>7.3 Helper tables .....</b>	<b>81</b>
7.3.1 Component long (table BWM.COMP_NAME_LONG) .....	81
7.3.2 Component long (table BWM.COMP_ATTR_LONG).....	82
<b>7.4 Exception tables .....</b>	<b>82</b>
<b>7.5 Incremental extraction .....</b>	<b>83</b>
 <b>8 Data mart schema information.....</b>	 <b>84</b>
<b>8.1 Data mart BWM Transaction Performance.....</b>	<b>84</b>
<b>8.2 Star schemas.....</b>	<b>84</b>
8.2.1 BWM hourly transaction performance transaction node star schema.....	84
8.2.1.1 Fact table BWM.F_TX_ND_HOUR .....	85
8.2.2 BWM daily transaction performance transaction node star schema .....	86
8.2.2.1 Fact table BWM.F_TX_ND_DAY .....	86
8.2.3 BWM weekly transaction performance transaction node star schema.....	87
8.2.3.1 Fact table BWM.F_TX_ND_WEEK .....	87
8.2.4 BWM monthly transaction performance transaction node star schema.....	88
8.2.4.1 Fact table BWM.F_TX_ND_MONTH .....	88
<b>8.3 Fact staging tables .....</b>	<b>89</b>
8.3.1 Fact staging table BWM.STG_TX_ND_HR .....	89
<b>8.4 Metric dimension tables .....</b>	<b>90</b>
8.4.1 BWM.D_TX_ND_METRIC .....	90
<b>8.5 Dimension tables.....</b>	<b>91</b>
8.5.1 Dimension table BWM.D_HOST .....	92
8.5.2 Dimension table BWM.D_TX.....	92
8.5.3 Dimension table BWM.D_TX_ND .....	92
8.5.4 Dimension table BWM.D_APP.....	93
<b>8.6 Mart translation tables.....</b>	<b>93</b>
8.6.1 BWM.T_TX_ND_METRIC.....	93
8.6.2 BWM.T_HOST .....	94
8.6.3 BWM.T_TX .....	94
8.6.4 BWM.T_TX_ND.....	95
8.6.5 BWM.T_APP .....	95
 <b>Notices .....</b>	 <b>98</b>

# 1 About this guide

This document describes the warehouse enablement pack, Version 5.2.0 for IBM® Tivoli Monitoring for Transaction Performance™ Version 5.2. This warehouse enablement pack (hereafter referred to as warehouse pack) is created for Tivoli Data Warehouse, Version 1.2.

This warehouse pack uses the product code of BWM. The script and process names for this warehouse pack include BWM and use BWM to define the schema owner of any tables created in the central data warehouse. 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

This warehouse pack allows you to 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 administrators and developers require the 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 used for reporting on transaction performance.

## 1.1 What's new in this warehouse pack

1. This version of the warehouse pack runs on Tivoli Data Warehouse, Version 1.2 only. Install this warehouse pack as a new warehouse pack or install over an existing version of the IBM Tivoli Monitoring for Transaction Performance warehouse pack.
2. New reports for this warehouse pack allow for additional reporting capability using the Tivoli Data Warehouse, Version 1.2 Crystal Reports.
3. Although you can install Tivoli Data Warehouse, Version 1.2 on z/OS systems, this warehouse pack cannot run on z/OS systems. You can run this warehouse pack on UNIX and Windows systems.
4. Support is available for multiple IBM Tivoli Monitoring for Transaction Performance database input sources. If you want to extract IBM Tivoli Monitoring for Transaction Performance data from a DB2 or Oracle database in one location and a DB2 or Oracle database in another location, specify all the data sources when you install this warehouse pack and data will be extracted from both sources when the extract processing runs.
5. The product code, BWM, is included in additional central data warehouse tables. This product code allows you to determine which data belongs to this warehouse pack.
6. Components whose component types have an MSrc\_Cd of 'MODEL1' are marked as SHARED for the MSrc\_Corr\_Cd column. This shows that the component is shared in the warehouse by any warehouse pack.
7. Warehouse packs for Tivoli Data Warehouse, Version 1.2 also provide the following new features:
  - A new directory structure
  - Script name changes
  - The data mart ETL automatically runs after a successful run of the central data warehouse ETL

## 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, 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:

- *IBM Tivoli Monitoring for Transaction Performance Installation Guide*, SC32-1385  
Provides prerequisite information and instructions for installing the Web Transaction Performance component. This guide also contains information that you might find useful after installing the product, such as uninstallation instructions and reference information about digital certificates.
- *IBM Tivoli Monitoring for Transaction Performance User's Guide*, SC32-1386  
Provides detailed procedures for using each of the Web Transaction Performance applications. The guide also describes the browser-based graphical user interface (GUI), the help system, and how to use Tivoli Decision Support to produce graphical reports from Web Transaction Performance data.
- *IBM Tivoli Monitoring for Transaction Performance Problem Determination Guide*, SC32-1387  
Provides the latest information about known product limitations and workarounds for the Web Transaction Performance component. To ensure that the information is the latest available, this document is provided only on the Web, where it is updated as needed.

## 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 10.

- *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 commands and lists troubleshooting tips for Tivoli Data Warehouse.
- *End-to-End e-business Transaction Management Made Easy*, SG24-6080  
Describes how to install, tailor, and configure the new IBM Tivoli Monitoring for Transaction Performance Version 5.2, which will assist you in determining the business performance of your e-business transactions in terms of responsiveness, performance, and availability.



### 1.3.3.2 IBM DB2, DB2 Data Warehouse Center, and DB2 Warehouse Manager library

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

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

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

The following DB2 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 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, 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 IBM DB2 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 Navigator 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

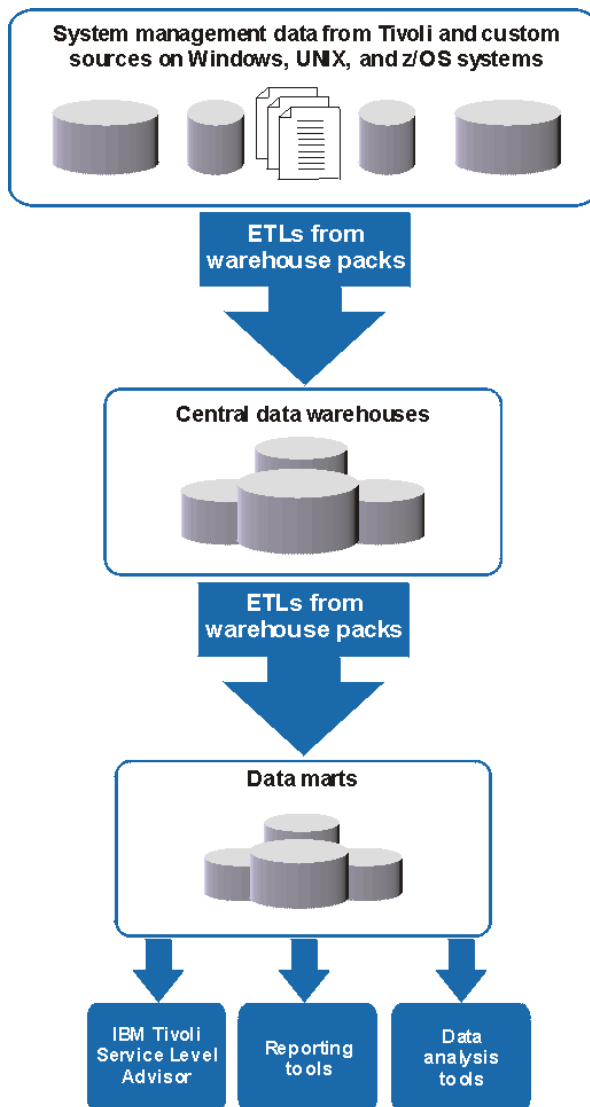
The following sections provide an overview of Tivoli Data Warehouse and the warehouse pack for IBM Tivoli Monitoring for Transaction Performance.

### 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 82.

## 2.2 Overview of IBM Tivoli Monitoring for Transaction Performance Version 5.2 warehouse pack

IBM Tivoli Monitoring for Transaction Performance, Version 5.2 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 or Oracle database products. This database, or databases, is regarded as the IBM Tivoli Monitoring for Transaction Performance, Version 5.2 *source databases* for this warehouse pack.

**\*\*If the warehouse pack reads from multiple TMTP source databases, ensure that the TMTP database sources monitor different management domains so that measurement data extracted from both databases does not overlap and skew the measurements.**

Once the IBM Tivoli Monitoring for Transaction Performance, Version 5.2 real time reporting data is stored in the source databases, the central data warehouse ETL process periodically (normally once a day) extracts data from the source database into the central data warehouse called TWH\_CDW. The central data warehouse database ETL process converts the data into the IBM Tivoli Monitoring for Transaction Performance, Version 5.2 warehouse pack data model shown in Figure 2. This data model allows the IBM Tivoli Monitoring for Transaction Performance, Version 5.2 reporting data to fit into the general schema of Tivoli Enterprise Data Warehouse, Version 1.2 for historical reporting and trending.

The Tivoli Monitoring for Transaction Performance source database tables that provide data to the TWH\_CDW are as shown in Figure 2

Table name	Table data uploaded into the central data warehouse
Host	HOST_ID, HOST_NAME, HOST_IP_ADDRESS
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

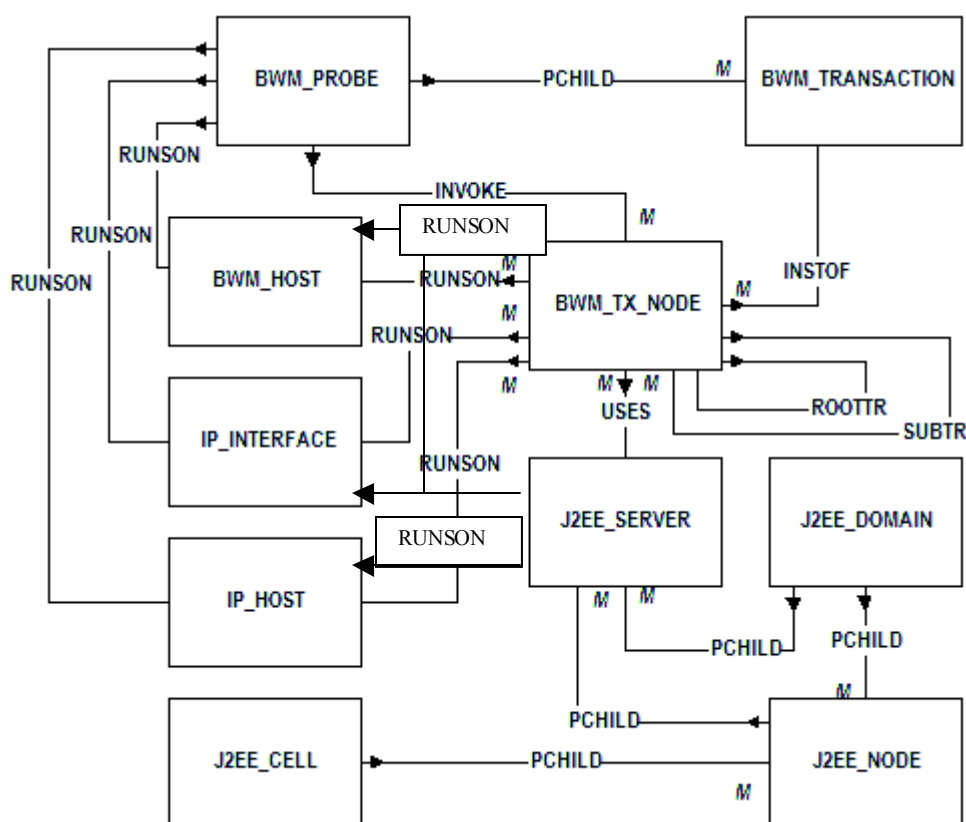
Table name	Table data uploaded into the central data warehouse
Aggregatedata	THRESH_VIOL_CNT, BAD_STATUS_CNT, ABORT_STATUS_CNT, COUNT, AVERAGE_VALUE, MAXIMUM_VALUE, MINIMUM_VALUE, AGGREGATE_ID, RELATIONMAP_ID, COLL_DATETIME, PARTIAL, PATT_TRANS_OM_ID, COLL_TIME_SPAN
Relationmap	RELATIONMAP_ID, MGMT_POLICY_OM_ID, PARENT_NODE_ID, CURRENT_NODE_ID, ROOT_NODE_ID, RELATED_NODE_ID
Patterntransaction	UUID1
Threshold	UUID1, THRPTINVERSE_UUID, THRESHOLDVALUE, TR_UUID
TR	UUID1, MANAGEMENTPOLICYID, RESPONSELEVEL
Managementpolicy	NAME, OBJECTVERSION, CREATOR, CREATED, DELETED, DELETOR, LASTUPDATED, UPDATOR, DESCRIPTION, UUID1, ISDELETE, TYPE1, STATE1, EDGEPOLICY_UUID, SCHEDULE_UUID, ENDPOINTGROUP_UUID

**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 5.2 warehouse pack component data model**

The following table contains a short description of the components used in IBM Tivoli Monitoring for Transaction Performance. For details on component relationships for IBM Tivoli Monitoring for Transaction Performance, see the Component relationship rule table on page 64.

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 J2EE data from WebLogic or WebSphere applications.
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).
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

Component name	Component type code	Description of component
		Server).
J2EE Node	J2EE_NODE	A J2EE node represents the machine upon which J2EE components run.
J2EE Domain	J2EE_DOMAIN	A J2EE domain represents the specified range of J2EE managed objects.
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.



### 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 tables that are used to create the reports
- A section that describes how the report input parameters work and the criteria for retrieving data for the report

Figure 4 shows the information for each predefined report.

Report name	Description	Table names
Average response time by application	This bar graph and detail report show individual application average response times across all hosts for the specified time period. Application transaction 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. The transaction response time measurement unit is seconds. 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 table used: BWM.F_TX_ND_DAY  Other mart tables used are: BWM.D_TX_ND_METRIC BWM.D_APP
Average response time by user	This bar graph and report show, at a high level, response times across all hosts, for each user, averaged over the time period set by an input parameter. User transaction response time is the average of all transactions' response times for that user for that time period. The users shown are determined by an input parameter. The transaction response time measurement unit is seconds. This report indicates the overall performance of user's transactions. This can aid in determining the overall transaction response time for many users to see if all users that run the same set of transactions experience the same kind of response time. This report utilizes the BWM Daily Transaction Node Star Schema.	Fact table used: BWM.F_TX_ND_DAY  Other mart tables used are: BWM.D_TX_ND_METRIC BWM.D_TX_ND
Average response time by policy	This bar graph and report show, at a high level, response times across all hosts, for each policy, averaged 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. 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 improved. This report utilizes the BWM Daily Transaction Node Star Schema.	Fact table used: BWM.F_TX_ND_DAY  Other mart tables used are: BWM.D_TX_ND_METRIC BWM.D_TX_ND

Report name	Description	Table names
Average response time by agent	This bar graph and detail report show transaction counts and response times for agents averaged over the desired time period. Input parameters determine the time period shown. Agent transaction response time is the average of all transactions' response times for that agent for the time period specified. The agents shown are also determined by an input parameter. The transaction response time measurement unit is seconds. This report demonstrates, at a high level, the performance of all transactions on an agent. It can help to determine whether transactions' overall response time has been due to any particular machine, which would indicate a resource issue. This report utilizes the BWM Daily Transaction Node Star Schema.	Fact table used: BWM.F_TX_ND_DAY  Other mart tables used are: BWM.D_TX_ND_METRIC BWM.D_HOST
Execution load by agent	This bar graph and detail report show transaction execution counts for each agent averaged over the desired time period. Input parameters determine the time period shown and the agents shown. The transaction counts include successful and failed transaction executions. This report indicates the load that each agent was under for a given timeframe. It can show unbalanced loads or application servers that need additional resources. This report utilizes the BWM Daily Transaction Node Star Schema.	Fact table used: BWM.F_TX_ND_DAY  Other mart tables used are: BWM.D_TX_ND_METRIC BWM.D_HOST
Execution load by application	This report shows the number of transactions that executed for a particular application across all hosts during the time period specified. Input parameters determine the time period shown and the applications shown. 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 Transaction Node Star Schema.	Fact table used: BWM.F_TX_ND_DAY  Other mart tables used are: BWM.D_TX_ND_METRIC BWM.D_APP
Execution load by user	This pie chart and detailed report show the transactions run by particular users over a given timeframe. Textual data shows the actual counts. These counts do not represent actual transaction counts unless the policy sampling rates of all transactions have been set to 100%. Otherwise, it shows proportionally how many transactions each user is running. Input parameters determine the time period shown and the users shown. The transaction counts include successful and failed transaction executions. This report indicates the transaction load that each user places on the systems running the transactions. This report utilizes the BWM Daily Transaction Node Star Schema.	Fact table used: BWM.F_TX_ND_DAY  Other mart tables used are: BWM.D_TX_ND_METRIC BWM.D_TX_ND
Execution load for policy <policy>	This line graph and detail report show the number of times that transactions, monitored by a given policy, have executed across all hosts during the time period specified. The time period and policy to show are specified by input parameters. The transaction counts include successful and failed transaction executions. This report gives an indication of the transaction load that a given policy	Fact table used: BWM.F_TX_ND_DAY  Other mart tables used

Report name	Description	Table names
	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.	are:  BWM.D_TX_ND_METRIC BWM.D_TX_ND
Transaction availability for <transaction>	This line graph shows the percentage availability of a transaction across its executing endpoints. 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.	Fact table used:  BWM.F_TX_ND_DAY  Other mart tables used are:  BWM.D_TX_ND_METRIC BWM.D_TX BWM.D_HOST
Overall transaction over time <transaction>	This line graph shows the average response time over time of a transaction specified by an input parameter over each agent, specified by an additional input parameter. This report indicates the relative performance of a transaction over each agent for comparison purposes. This report utilizes the BWM Daily Transaction Node Star Schema.	Fact table used:  BWM.F_TX_ND_DAY  Other mart tables used are:  BWM.D_TX_ND_METRIC BWM.D_TX BWM.D_HOST
Slowest transactions	This extreme case report shows the top N edge transactions that were the slowest transactions over the time period specified for the specified transactions, users, applications and hosts. The time period, transactions, number of transactions for top N, users, applications and agents to show are specified by input parameters. 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.	Fact table used:  BWM.F_TX_ND_DAY  Other mart tables used are:  BWM.D_TX_ND_METRIC BWM.D_TX BWM.D_HOST BWM.D_TX_ND BWM.D_APP
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. 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 table used:  BWM.F_TX_ND_DAY  Other mart tables used are:  BWM.D_TX_ND_METRIC BWM.D_HOST BWM.D_TX_ND
Slowest throughput	This report shows the top N transactions with the slowest throughput. Throughput is defined as the number of transactions executed per day for the specified time period for values for	Fact table used:  BWM.F_TX_ND_DAY

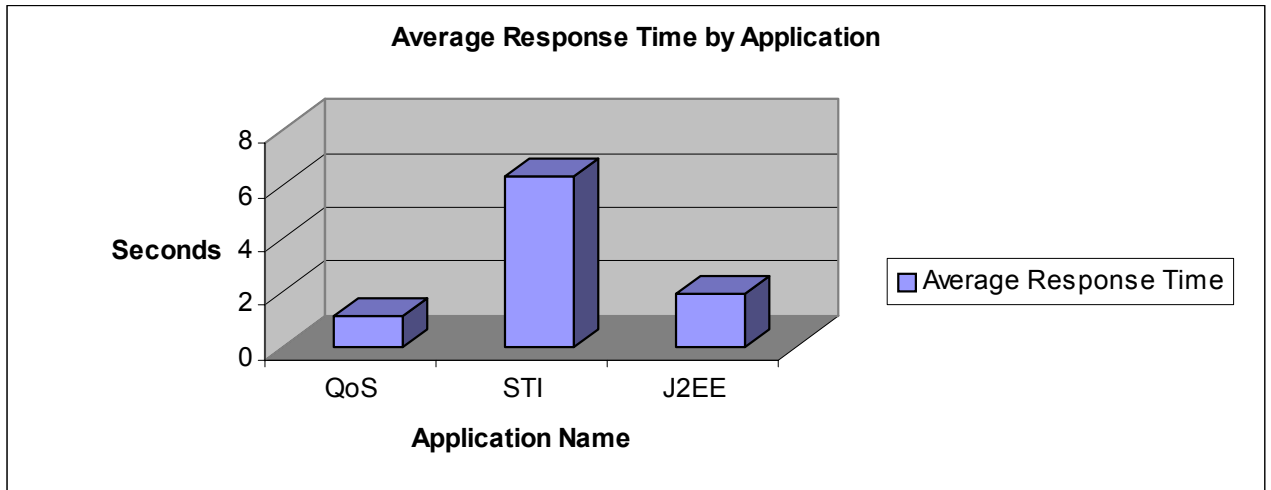
Report name	Description	Table names
	transaction, user, agent and application. The time period, transactions, number of transactions for top N, users, applications and agents to show are specified by input parameters. This report shows the least number of transactions being executed for a set of transactions, users, applications and hosts. This report utilizes the BWM Daily Transaction Node Star Schema.	Other mart tables used are: BWM.D_TX_ND_METRIC BWM.D_TX BWM.D_HOST BWM.D_TX_ND BWM.D_APP

**Figure 4. IBM Tivoli Monitoring for Transaction Performance, Version 5.2 warehouse pack  
Predefined reports**

### 3.1 Sample report layouts

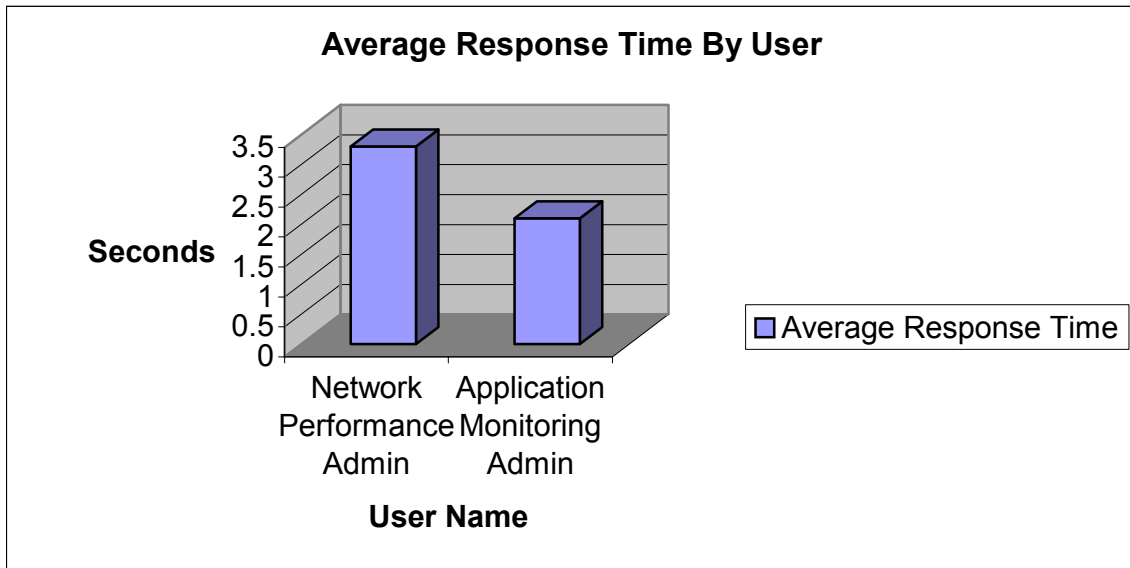
The following are samples of the reports provided.

#### 3.1.1 Average Response Time by Application



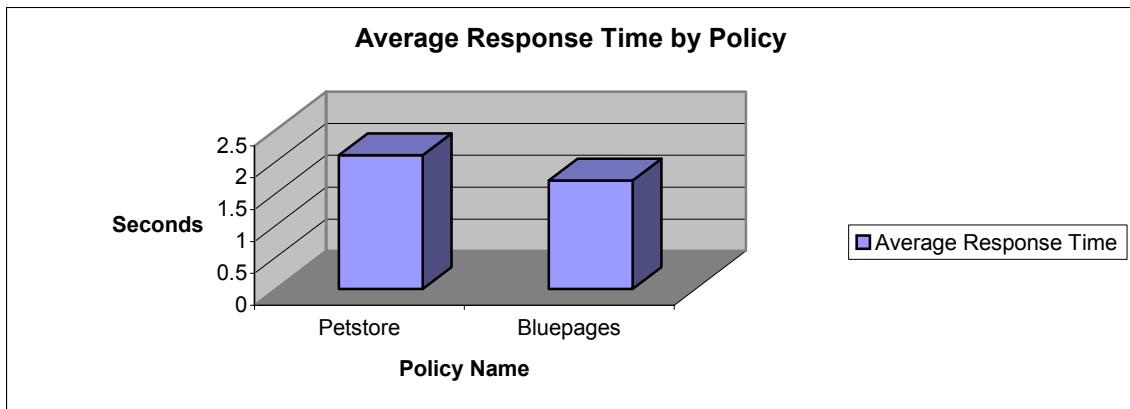
Average Response Time by Application	
<u>Application Name</u>	<u>Average Response Time (Seconds)</u>
J2EE	1.9
QoS	1.1
STI	6.3

### 3.1.2 Average Response Time by User



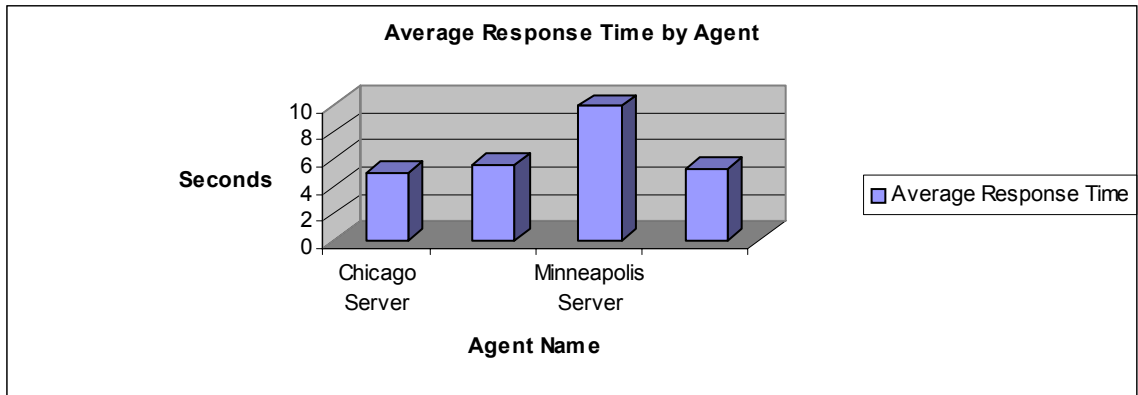
Average Response Time by User	
<u>User Name</u>	<u>Average Response Time (Seconds)</u>
Application Monitoring Admin	2.1
Network Performance Admin	3.3

### 3.1.3 Average Response Time by Policy



Average Response Time by Policy	
<u>Policy Name</u>	<u>Average Response Time (Seconds)</u>
Bluepages	2.1
Petstore	1.7

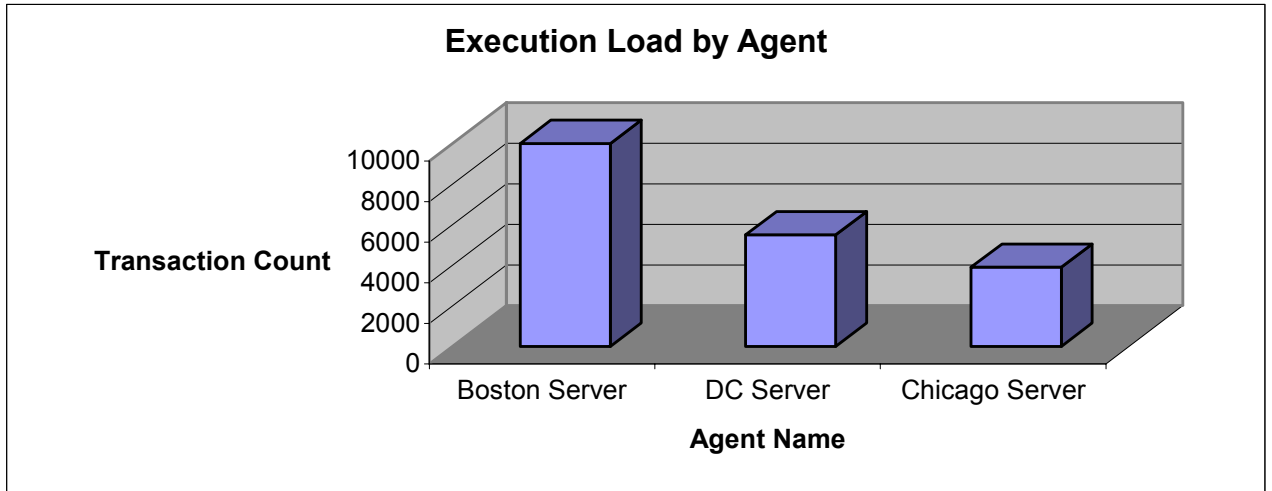
### 3.1.4 Average Response Time by Agent



Average Response Time by Agent

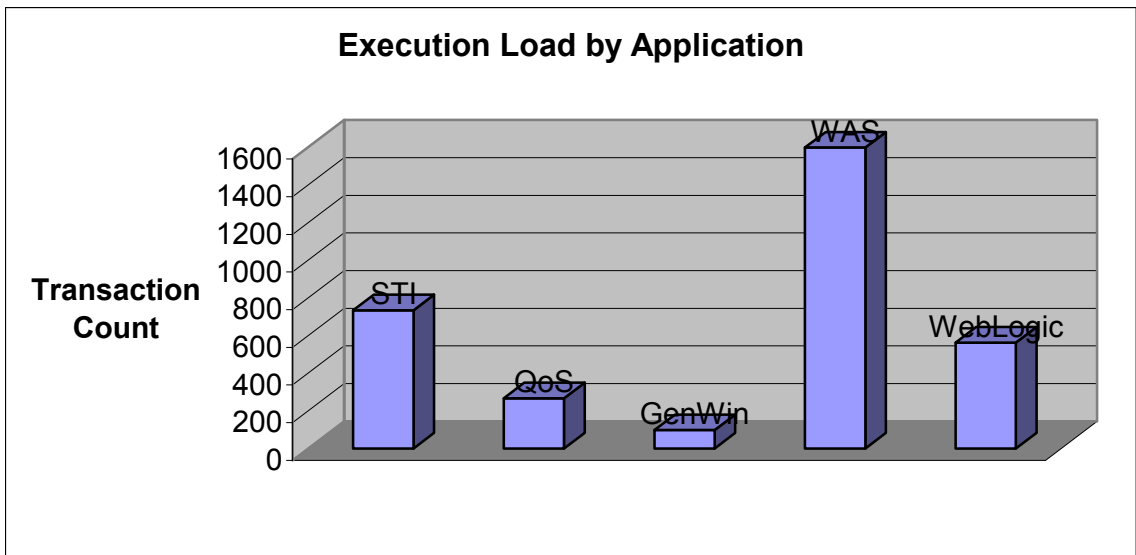
<u>Agent Name</u>	<u>Average Response Time (Seconds)</u>
Chicago Server	5.1
Detroit Server	5.5
Milwaukee Server	5.9
Minneapolis Server	7.6

### 3.1.5 Execution Load by Agent



Execution Load by Agent	
<u>Agent Name</u>	<u>Transaction Count</u>
Boston Server	10,000
DC Server	5,500
Chicago Server	3,900
<b>Total</b>	<b>19,400</b>

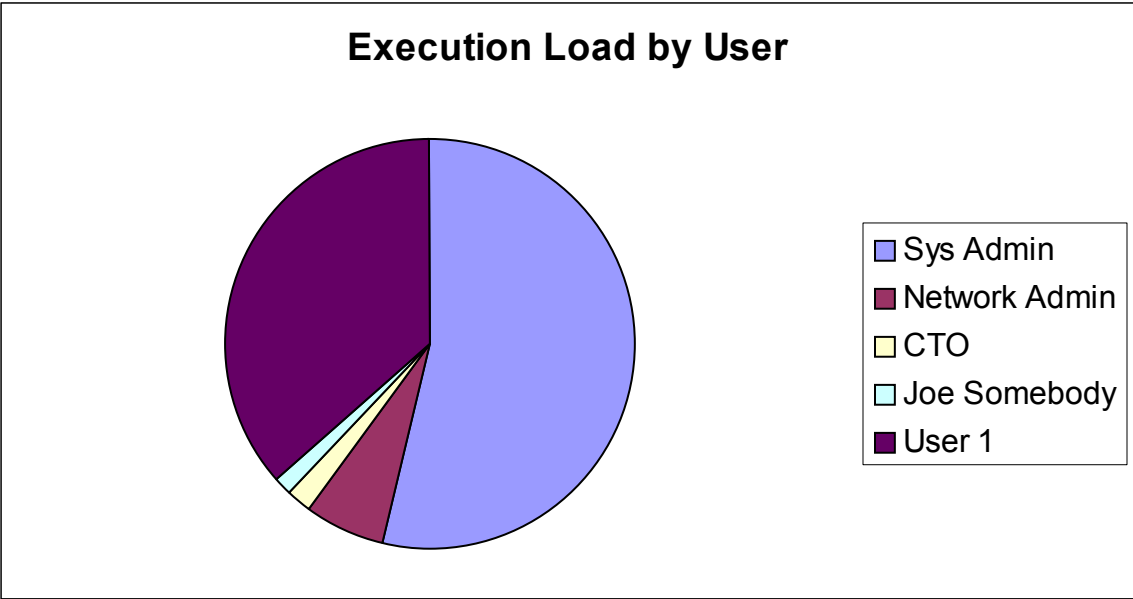
### 3.1.6 Execution Load by Application



Execution Load by Application	
<u>Application Name</u>	<u>Transaction Count</u>
WAS	1,599
STI	735
WebLogic	563
QoS	267
GenWin	98
<b>Total</b>	<b>3,262</b>

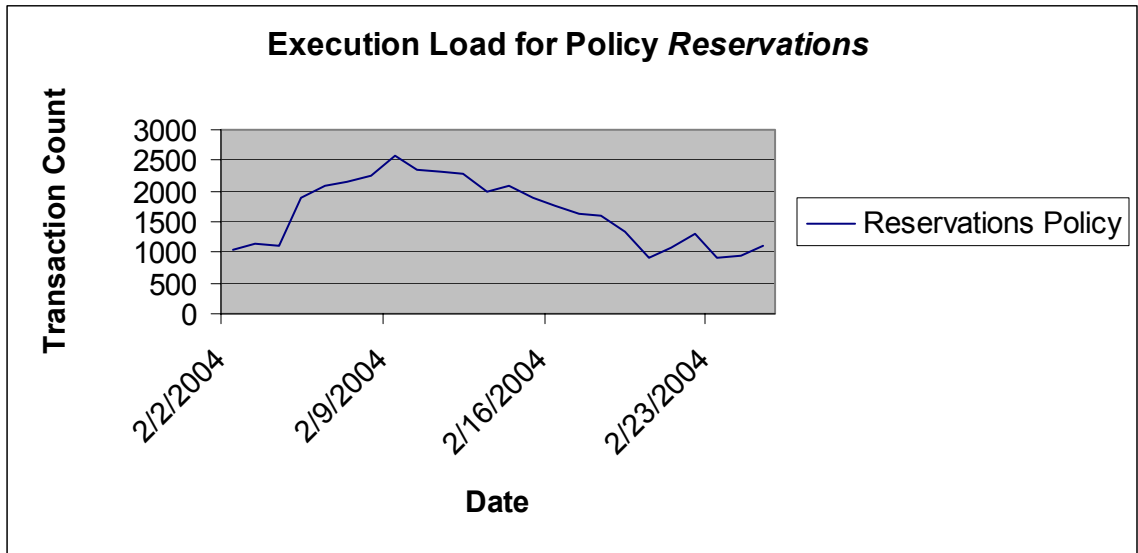


### 3.1.7 Execution Load by User



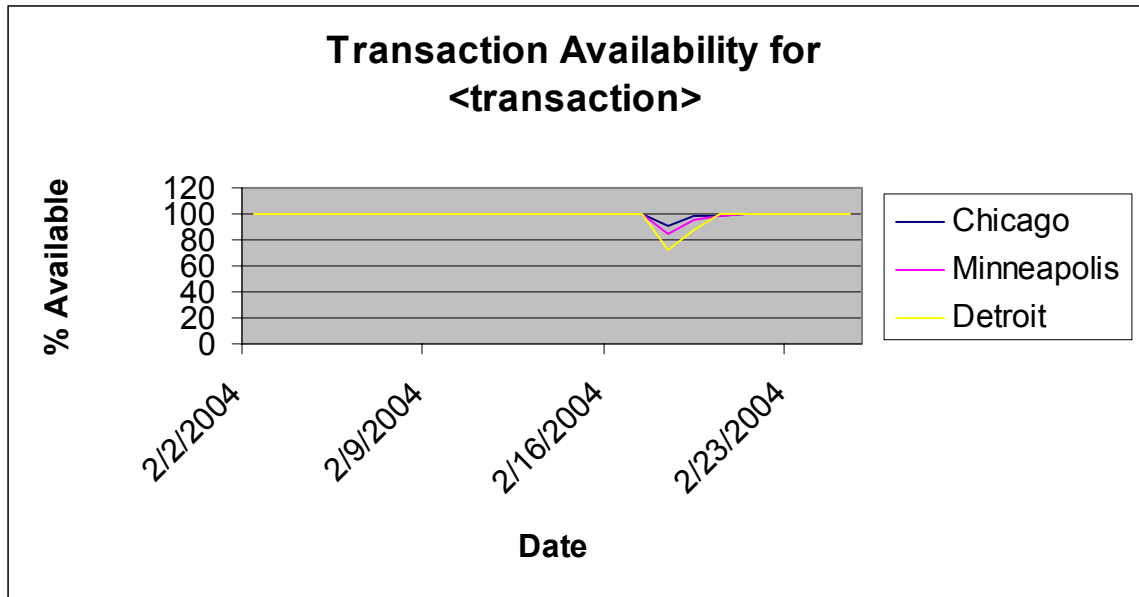
Execution Load by User	
User Name	Transaction Count
Sys Admin	500
User 1	305
Network Admin	130
CTO	25
Joe Somebody	10
Total	970

### 3.1.8 Execution Load For Policy <policy>



Execution Load for Policy <i>Reservations</i>	
Date	Transaction Count
2/2/2004	1032
2/4/2004	1132
2/6/2004	1109
2/8/2004	1905
2/10/2004	2103
Total	7281

### 3.1.9 Transaction Availability for <transaction>



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 days shown	100%

#### Agent Name

Detroit

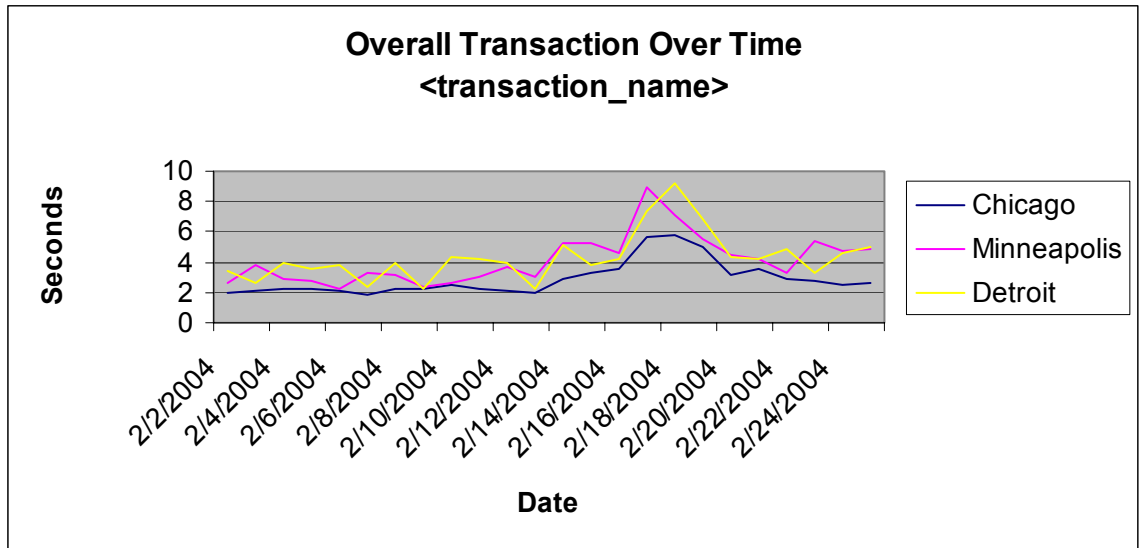
Date	% Available
2/2/2004	100%
2/3/2004	100%
2/4/2004	100%
2/5/2004	100%
2/6/2004	100%
Average availability over days shown	100%

#### 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 days shown	100%

### 3.1.10 Overall Transaction Over Time <transaction>



**Overall Transaction Over Time <transaction>**

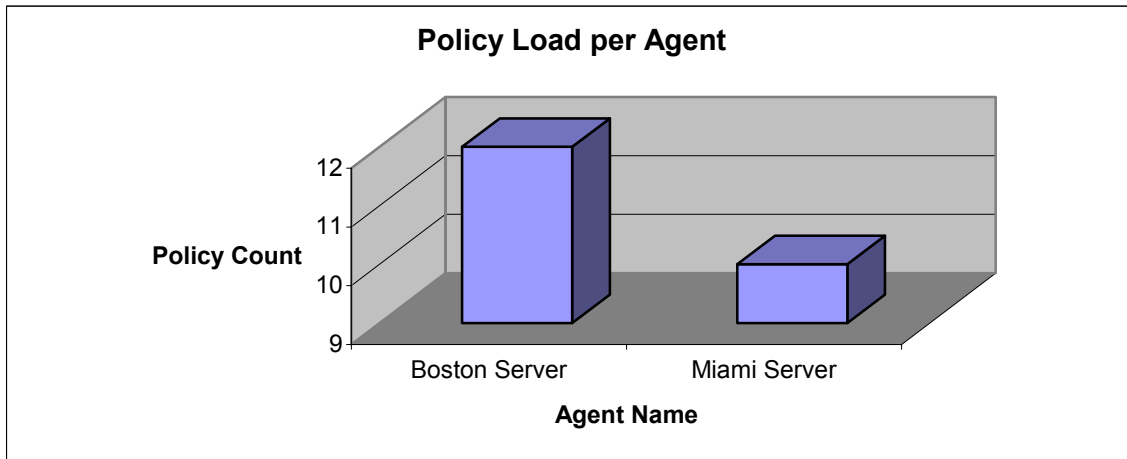
<b>Agent Name</b>	
<b>Chicago</b>	
<b>Date</b>	<b>Average Response Time</b>
2/2/2004	2.1
2/4/2004	1.9
2/6/2004	2.0
2/8/2004	2.2
2/10/2004	2.1
<b>Average over period</b>	<b>2.1</b>
<b>Agent Name</b>	
<b>Detroit</b>	
<b>Date</b>	<b>Average Response Time</b>
2/2/2004	4.0
2/4/2004	2.2
2/6/2004	4.1
2/8/2004	2.1
2/10/2004	3.0
<b>Average over days shown</b>	<b>3.1</b>
<b>Agent Name</b>	
<b>Minneapolis</b>	
<b>Date</b>	<b>Average Response Time</b>
2/2/2004	3.1
2/4/2004	2.0
2/6/2004	3.9
2/8/2004	3.7
2/10/2004	3.5
<b>Average over days shown</b>	<b>3.3</b>

### 3.1.11 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	Transaction A	NetPerfAdmin	Chicago Server	5.6	0.75	15.6	2004-02-02
STI	Transaction B	AppAdmin	Chicago Server	5.1	0.99	10.1	2004-02-03

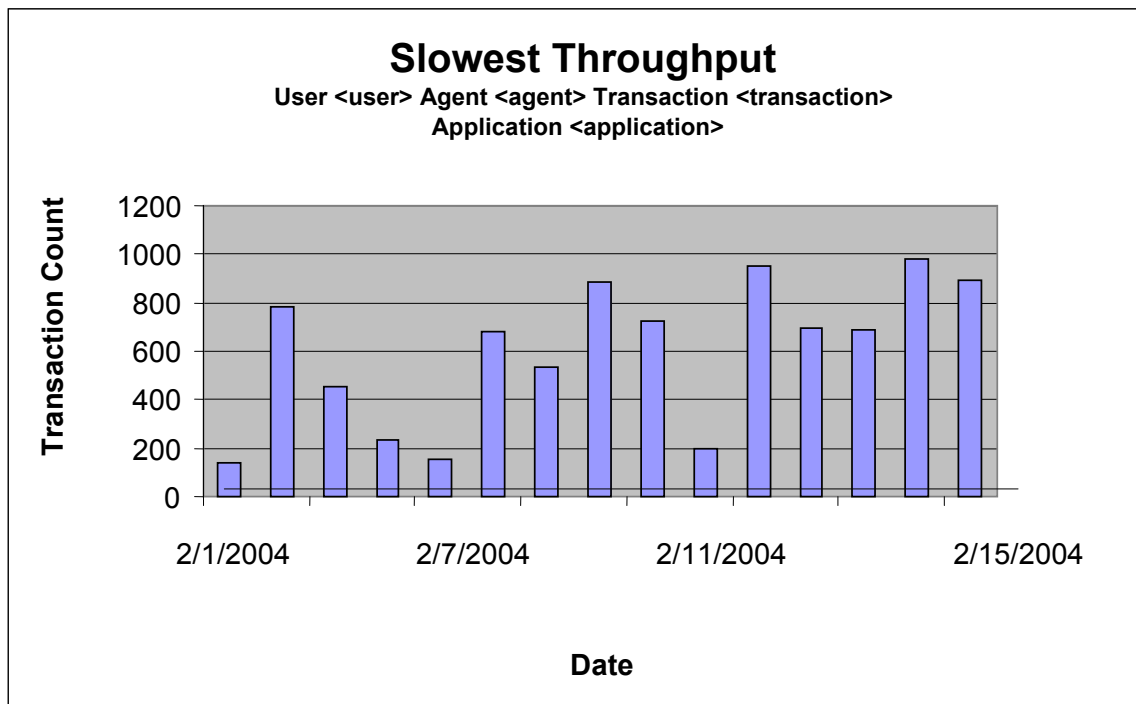
### 3.1.12 Policy Load per Agent



Policy Load per Agent

<u>Agent Name</u>	<u>Policy Count</u>
Boston Server	12
Chicago Server	15
Los Angeles Server	5
Miami Server	10
Seattle Server	4
<b>Average Policy Load</b>	<b>9.2</b>

### 3.1.13 Slowest Throughput



Slowest Throughput					
Application	Transaction	User	Agent	Transaction Count	Transaction Start Date
WAS	Transaction A	NetPerf Admin	Chicago Server	560	2004-02-02
STI	Transaction B	App Admin	Chicago Server	350	2004-02-03

### 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 hit the 'Select' button so the selection will appear in the box below of choices for that parameter. After all parameters have been entered and selected, hit the 'OK' button 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 bring pop up a calendar from which can you point and click to select the dates.

Note: 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 and the report's graph and data show on the second and subsequent pages. Look for the '1+' indicator at the top of the first report page to see that there are additional report pages.

### 3.2.1 Average Response Time by Application

For this report, the average response time is summed 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 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. The default value for the Start\_Date is yesterday and a week before yesterday for the End\_Date. The date values are chosen from a pop-up calendar if they are to be changed from the default values.

#### 3.2.1.2 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\*', etc. 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 this report, the average response time is summed over the time period requested and then grouped by the invoking user, showing only edge transactions. 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 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. The default value for the Start\_Date is yesterday and a week before yesterday for the End\_Date. The date values are chosen from a pop-up calendar if they are to be changed from the default values.

#### 3.2.2.2 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\*', etc. 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 to show.

### 3.2.3 Average Response Time by Policy

For this report, the average response time is summed 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 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. The default value for the Start\_Date is yesterday and a week before yesterday for the End\_Date. The date values are chosen from a pop-up calendar if they are to be changed from the default values.

### 3.2.3.2 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\*', etc. 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 to show.

### 3.2.4 Average Response Time by Agent

For this report, the average response time is summed over the time period requested and then grouped by host name. 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 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. The default value for the Start\_Date is yesterday and a week before yesterday for the End\_Date. The date values are chosen from a pop-up calendar if they are to be changed from the default values.

#### 3.2.4.2 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\*', etc. 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 to show.

### 3.2.5 Execution Load by Agent

For this report, the transaction execution count is summed over the time period requested and then grouped by host name. 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:

$$\text{sum}(\text{fact\_table.sample\_count})$$

#### 3.2.5.1 Start Date and End Date parameters

The Start\_Date and End\_Date input parameter values are chosen 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. The default value for the Start\_Date is yesterday and a week before yesterday for the End\_Date. The date values are chosen from a pop-up calendar if they are to be changed from the default values.



### 3.2.5.2 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\*', etc. 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 to show.

## 3.2.6 Execution Load by Application

For this report, the transaction execution count is summed 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)
```

### 3.2.6.1 Start Date and End Date parameters

The Start\_Date and End\_Date input parameter values are chosen 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. The default value for the Start\_Date is yesterday and a week before yesterday for the End\_Date. The date values are chosen from a pop-up calendar if they are to be changed from the default values.

### 3.2.6.2 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\*', etc. This input parameter is a text field and only characters should be entered for the application name.

## 3.2.7 Execution Load by User

For this report, the transaction execution count is summed over the time period requested and then grouped by the invoking user, showing only edge transactions. 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)
```

### 3.2.7.1 Start Date and End Date parameters

The Start\_Date and End\_Date input parameter values are chosen 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. The default value for the Start\_Date is yesterday and a week before yesterday for the End\_Date. The date values are chosen from a pop-up calendar if they are to be changed from the default values.

### 3.2.7.2 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\*', etc. 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 to show.

### 3.2.8 Execution Load For Policy <policy>

For this report, the transaction execution count is summed over the time period requested and then grouped by 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:

$\text{sum}(\text{fact\_table.sample\_count})$

#### 3.2.8.1 Start Date and End Date parameters

The Start\_Date and End\_Date input parameter values are chosen 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. The default value for the Start\_Date is yesterday and a week before yesterday for the End\_Date. The date values are chosen from a pop-up calendar if they are to be changed from the default values.

#### 3.2.8.2 Policy Name

The policy name input parameter should be entered as a single, exact value. The graph shows the transactions run for the one policy.

### 3.2.9 Transaction Availability For <transaction>

For this report, the transaction availability is calculated 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:

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

This translates to the following fact table columns:

$\text{sum}(\text{fact\_table.sample\_count}) - \text{sum}(\text{fact\_table.error\_count})$

#### 3.2.9.1 Start Date and End Date parameters

The Start\_Date and End\_Date input parameter values are chosen 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. The default value for the Start\_Date is yesterday and a week before yesterday for the End\_Date. The date values are chosen from a pop-up calendar if they are to be changed from the default values.

#### 3.2.9.2 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\*', etc. 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 to show.

#### 3.2.9.3 Transaction Name

The transaction name input parameter should be entered as a single, exact value. The graph shows the transaction's percent available for the agents selected.

### 3.2.10 Overall Transaction Over Time <transaction>

For this report, the transaction average response time 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:

$$\text{sum}(\text{fact\_table.sample\_count} * \text{fact\_table.sample\_count}) / \text{sum}(\text{fact\_table.sample\_count})$$

#### 3.2.10.1 Start Date and End Date parameters

The Start\_Date and End\_Date input parameter values are chosen 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. The default value for the Start\_Date is yesterday and a week before yesterday for the End\_Date. The date values are chosen from a pop-up calendar if they are to be changed from the default values.

#### 3.2.10.2 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\*', etc. 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 to show.

#### 3.2.10.3 Transaction Name

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

### 3.2.11 Policy Load per Agent

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

$$\text{count}(\text{distinct management\_policy})$$

This translates to the following D\_TX\_ND dimension table column:

$$\text{count}(\text{distinct management\_policy})$$

#### 3.2.11.1 Start Date and End Date parameters

The Start\_Date and End\_Date input parameter values are chosen 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. The default value for the Start\_Date is yesterday and a week before yesterday for the End\_Date. The date values are chosen from a pop-up calendar if they are to be changed from the default values.

#### 3.2.11.2 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\*', etc. 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 to show.

## 3.2.12 Slowest Throughput

For this report, the transaction execution count is summed over the time period requested and then grouped by application, invoking user, host name, transaction name and date. 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)`

### 3.2.12.1 Start Date and End Date parameters

The Start\_Date and End\_Date input parameter values are chosen 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. The default value for the Start\_Date is yesterday and a week before yesterday for the End\_Date. The date values are chosen from a pop-up calendar if they are to be changed from the default values.

### 3.2.12.2 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\*', etc. 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 to show.

### 3.2.12.3 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\*', etc. This input parameter is a text field and only characters should be entered for the application name.

### 3.2.12.4 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\*', etc. 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 to show.

### 3.2.12.5 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.

## 3.2.13 Slowest Transactions

For this report, the transaction average response time is shown over the time period requested and then grouped by application, invoking user, host name, transaction name and date.

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

`sum(avg_response_time * transaction_execution_count) / sum(transaction_execution_count)`

This translates to the following fact table columns:

`sum(fact_table.sample_count * fact_table.sample_count) / sum(fact_table.sample_count)`

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

```
min(avg_response_time)
```

This translates to the following fact table columns:

```
min(fact_table.min_value)
```

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

```
max(avg_response_time)
```

This translates to the following fact table columns:

```
max(fact_table.max_value).
```

### **3.2.13.1 Start Date and End Date parameters**

The Start\_Date and End\_Date input parameter values are chosen 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. The default value for the Start\_Date is yesterday and a week before yesterday for the End\_Date. The date values are chosen from a pop-up calendar if they are to be changed from the default values.

### **3.2.13.2 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\*', etc. 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 to show.

### **3.2.13.3 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\*', etc. This input parameter is a text field and only characters should be entered for the application name.

### **3.2.13.4 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\*', etc. 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 to show.

### **3.2.13.5 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\*', etc. 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 to show.

### **3.2.13.6 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 highest average response time will be shown in descending order.

## 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 5.2.0, you must install the following software:

- IBM Tivoli Monitoring for Transaction Performance, Version 5.2
- IBM DB2 Universal Database, Version 7.2
- Fix pack 8e, 9, 10, or 10a for IBM DB2 Universal Database, Version 7.2.
- 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 ADAPT00281799 (See section 4.1.1 below for directions on how to install)
- (Optional) IBM Tivoli Service Level Advisor™ 1.2.1 Fix Pack 1
- If the IBM Tivoli Monitoring for Transaction Performance warehouse pack 5.1.0 is installed, it must be upgraded to version 5.1.0.5 and the BWM\_c05\_Upgrade51\_Process must be run to upgrade the warehouse pack schema to work with IBM Tivoli Monitoring for Transaction Performance 5.2 before upgrading to this warehouse pack.

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 5.2 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

To view the defects that this patch fixes, see the following Web site:

<http://support.businessobjects.com/library/kbase/articles/c2015312.asp>.

Install the patch for Crystal Enterprise 9:

- a. Access Crystal product support from the following Web site:  
<http://support.businessobjects.com/library/kbase/articles/c2015312.asp>
- b. From this general support location, select Crystal Enterprise Version 9.
- c. Choose or create a temporary location for the patch zip file.
- d. Transfer the patch to your Crystal Enterprise Server system using FTP from the following Web site: [ftp://ftp.crystaldecisions.com/outgoing/EHF/ce90comwin\\_en.zip](ftp://ftp.crystaldecisions.com/outgoing/EHF/ce90comwin_en.zip).  
You can also access translated versions of this patch from this Web site.
- e. Unzip the file into the same temporary location and enter this command under **Start > Run:**  
**ce90comwin\_en.exe.**
- f. The Crystal installation wizard installs the patch when the process is complete.
- g. Re-IPL your Crystal Enterprise Server.

## **4.2 Product notes and limitations**

### **4.2.1 Product notes**

The warehouse pack for IBM Tivoli Monitoring for Transaction Performance 5.2 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.

#### **4.2.2.2 Reinstalling the IBM 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 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 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 Warehouse `CDW_c05_Prune_and_Mark_Active` process.
4. Reinstall the IBM Tivoli Monitoring for Transaction Performance warehouse pack.

**Note:** The IBM Tivoli Monitoring for Transaction Performance warehouse pack retrieves all the data from the IBM Tivoli Monitoring for Transaction Performance database. If you want a lesser amount of data in the warehouse, you must run the prune process after the install to remove the unwanted measurements. This leaves the component data that the central data warehouse needs.

#### **4.2.2.3 Initial Data Extraction**

The IBM Tivoli Monitoring for Transaction Performance warehouse pack retrieves all the data from the IBM Tivoli Monitoring for Transaction Performance database during the first extract processing. The last step of the data mart ETL process prunes the data to the values set in the `TWG.Prune_Msmt_Control` and `BWM.Prune_Mart_Control` tables and eliminates any older data.



#### 4.2.2.4 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 rerun 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 warehouse user needs to update the extract control information to pick up data from the point of failure and rerun the central data warehouse ETL.

If you gather large amounts of data daily or the problem was left unattended for a long time, re-extracting the data missed during the central data warehouse ETL failures can take a significant amount of time and data resources. If the IBM Tivoli Monitoring for Transaction Performance source database had data purged after the central data warehouse ETL failure occurred, but before the failure was corrected and the central data warehouse ETL reruns, then the re-extraction of data will not 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.

**Step 1:** Review the times and sequences collected for each run of the warehouse. From the DB2 command line processor, run the Show\_ExtLog\_Values.sql script in the misc/tools directory or run the following DB2 command after connecting to the TWH\_CDW database as a valid and authorized DB2 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'
```

The following is sample output from the 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:** View the date for which the initial failure occurred and view the Extract Logs value in the ExtLog\_FROM\_INTSEQ column.

**Step 3:** Update the Extract\_Control tables from the ExtCtl\_FROM\_INTSEQ integer sequence column, with the value from the ExtLog\_FROM\_INTSEQ Extract Logs column that you selected in step 2. Use the following SQL statement to update the Extract Control table.

Note: The “?” represents the selected sequence number.

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

It is not necessary to update the Extract Control tables ExtCtl\_TO\_INTSEQ column. The central data warehouse ETL 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 5.2.0 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

If the IBM Tivoli Monitoring for Transaction Performance warehouse pack, Version 5.1.0 is installed, you must upgrade to Version 5.1.0.5. First, do the following:

- Install any available patches for the 5.1.0.5 Version
- Run the BWM\_c05\_Upgrade51\_Process process

#### 4.4.1 Tivoli Data Warehouse, Version 1.2 requirement

This warehouse pack only runs on Tivoli Data Warehouse, Version 1.2. You must install or upgrade the Tivoli Data Warehouse to Version 1.2 Fix Pack 2 before you install or upgrade to Version 5.2 of the IBM Tivoli Monitoring for Transaction Performance warehouse pack. See *Installing and Configuring Tivoli Data Warehouse, Version 1.2* for details.

When you upgrade to Tivoli Data Warehouse, Version 1.2, all warehouse packs previously installed for Tivoli Enterprise Data Warehouse, Version 1.1 are automatically upgraded to the Version 1.2 format. These previously installed warehouse packs only have the Tivoli Enterprise Data Warehouse, Version 1.1 functionality. Required changes to make the warehouse pack fully functional take place when you install the warehouse pack for Version 1.2.

#### 4.4.2 Creating warehouse database backups

Before installing or upgrading to a new warehouse pack, make backups of the TWH\_CDW, TWH\_MD, and TWH\_MART warehouse databases as well as the \$TWH\_TOPDIR/apps directory. This allows you to establish a point of recovery in case of installation failures.

### 4.4.3 Upgrading to Version 5.2.0 of the IBM Tivoli Monitoring for Transaction Performance warehouse pack

When you install this warehouse pack, the install process renames the existing fact tables and creates new tables. Once the new fact tables are created, the data from the renamed tables is copied, a table at a time, into the new fact tables so no data is lost. Because the fact tables can contain large amounts of data, you must accurately size the transaction logs to enable the data to be inserted from the old fact 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 fact tables, all of the logs will work because automatic commits occur after each SQL statement.

When the install runs, control tables that keep track of which SQL statements have already run are created. When the install process is run more than one time these control tables ensure that SQL statements that have already run successfully are not rerun. If the transaction logs are not sized correctly and cause the install process to fail, resize them, add more or both, and then run the upgrade install again.

### 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 control server or remote agent site where you have scheduled the ETL step to run. For example, if the Tivoli Monitoring for Transaction Performance database is implemented on an Oracle database server, then you must install the Oracle client on the control server or remote agent site. This allows the ODBC connection to use the client to communicate with the Tivoli Monitoring for Transaction Performance source database.

### 4.4.5 ODBC drivers

See *Installing and Configuring Tivoli Data Warehouse, Version 1.2* 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 Oracle Version 8 can also be used with Oracle Version 9. The driver used with DB2 Version 7 can also be used with DB2 Version 8.

TMTF\_DB\_SRC is the default Tivoli Monitoring for Transaction Performance ODBC data source name in the BWM\_TMTF\_DB\_SRC\_Source warehouse source setting. You can specify additional Tivoli Monitoring for Transaction Performance ODBC data sources to extract data from multiple databases. Specify the additional Tivoli Monitoring for Transaction Performance ODBC databases when you install the warehouse pack or by running the twh\_configwep warehouse command. See *Installing and Configuring Tivoli Data Warehouse, Version 1.2* for more information on the usage of this command.

## 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 IBM 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*.

ODBC source	User ID	Password	Database type	Server name
-------------	---------	----------	---------------	-------------

ODBC source	User ID	Password	Database type	Server name
The default data source name for the ODBC connection is TMTP_DB_SRC.	<UserID>  This is the user id to access the TMTP database using the ODBC connection	<Password>  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 (twh\_install\_props.cfg file).

Note: Do not uninstall and reinstall your Tivoli Data Warehouse Enablement Pack environment until you read and understand the following critical scenario.

If you installed Tivoli Data Warehouse Enablement Pack, Version 5.2 in a Warehouse environment against an IBM Tivoli Monitoring for Transaction Performance source database and ran ETL processes and generated reports, you must perform the following steps before uninstalling the Tivoli Data Warehouse Enablement Pack with the intention to reinstall it pointing to the same source database:

- 1.) Locate the bwm\_cdw\_del\_data.sql file in the following directory: <Tivoli Warehouse install directory>/apps/bwm/pkg/v510/cdw/dml. This SQL file executes during the uninstallation.
- 2.) Comment out the following lines by appending '--' at the beginning of each line:
  - delete from twg.extract\_control where extctl\_target like 'BWM%';
  - delete from twg.extract\_log where extlog\_target like 'BWM%';
- 3.) Save the file.
- 4.) Run the uninstallation for the Tivoli Data Warehouse Enablement Pack to maintain the extract control values for the Tivoli Data Warehouse Enablement Pack.

After performing the above steps, you can now reinstall the Tivoli Data Warehouse Enablement Pack against the same source database. The ETL process will reengage from where it stopped from the extract control.

## 4.6 Post-installation procedures

### 4.6.1 Scheduling warehouse pack processes

The warehouse pack extract schedule is set during the installation of the warehouse pack. The data mart ETL extract runs automatically after the central data warehouse ETL successfully completes. You can reschedule the warehouse pack extract schedule after you install the warehouse pack. See *Installing and Configuring Tivoli Data Warehouse* for the procedure to schedule the ETLs. Use these process dependencies when changing any extract scheduling:

- Initialization process** none
- Process dependencies**
  - Located in the BWM\_Tivoli\_Monitoring\_for\_Transaction\_Performance\_v5.2.0\_Subject\_Area subject area
  - The processes should be run in the following order:
    1. BWM\_c10\_CDW\_Process

### 4.6.2 Change Prune Control values from installed defaults

The prune control values are configured with default values when you install the warehouse pack. Msmt data is pruned after it has been in the warehouse more than 3 months. Hourly and daily mart data is pruned after it has been in the data mart more than 3 months. 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 the prune control values.

### 4.6.3 Accessing the application source databases

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 5.2 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 warehouse source, you must verify that the IBM Tivoli Monitoring for Transaction Performance database exists and that you can connect to it. For DB2, 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.

By default, the ODBC data source name assigned to the IBM Tivoli Monitoring for Transaction Performance source database after the warehouse pack installation is TMTP\_DB\_SRC. Before you run the ETL processes, ensure that the warehouse sources and the warehouse targets specific to the IBM Tivoli Monitoring for Transaction Performance application are correctly defined. These sources must include username, password, and system data source.

### 4.6.4 Configuring warehouse source and target settings

The following sources and targets are created by the IBM Tivoli Monitoring for Transaction Performance, Version 5.2.0 warehouse pack:

- BWM\_TWH\_CDW\_Source: a source object for the TWH\_CDW database
- 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 TWH\_CDW database
- BWM\_TWH\_MART\_Target: a target for the TWH\_MART database

The warehouse sources and target values are already set when Tivoli Data Warehouse, Version 1.2, installs the warehouse pack. 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.
  - 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.
- Do not change the value of the Data Source field. It must be TWH\_CDW or TWH\_CDWn.
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. The default value is *db2admin*.
    - In the Password field, type the password used to access the data mart.
    - Do not change the value of the Data Source field. It must be TWH\_MART or TWH\_MARTn.
  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. The default value is *db2admin*.
    - In the Password field, type the password used to access the central data warehouse.
    - Do not change the value of the Data Source field. It must be TWH\_CDW or TWH\_CDWn.
  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. The default value is *db2admin*.
    - In the Password field, type the password used to access the data mart.
    - Do not change the value of the Data Source field. It must be TWH\_MART or TWH\_MARTn.

## 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:

- Reports

### Changed objects:

- Prior reports have been redone in Crystal Reports

### Removed objects:

- BWM\_c05\_Upgrade51\_Process
- BWM\_TWH\_MD\_Target

## 4.8 Un-installation of the warehouse pack

Perform the following steps to uninstall the warehouse pack:

1. Uninstall the warehouse pack as described in *Installing and Configuring Tivoli Data Warehouse*.

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.STG\_HOST  
BWM.STG\_APPLICATION  
BWM.STG\_USER  
BWM.STG\_TRANSACTION  
BWM.STG\_NODE  
BWM.STG\_RELATIONMAP  
BWM.STG\_AGGREG\_DATA  
BWM.STG\_MGMTPOLICY  
BWM.STG\_PATTERN  
BWM.STG\_THRESHOLD  
BWM.STG\_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 *Installing and Configuring Tivoli Data Warehouse* and *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 *Installing and Configuring Tivoli Data Warehouse* and *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.

Run the following command to back up a database from the DB2 Command Line Processor:

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

You can also use the DB2 Control Center to back up databases.

See *Installing and Configuring Tivoli Data Warehouse, Version 1.2* for details on 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 pruned 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. You can customize the time and interval schedule to run daily, weekly, and monthly.

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 pruned 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'
```

## 5.2.2 Data mart

Pruning data from the data mart fact tables is implemented in the BWM\_m05\_s050\_mart\_prune process. The prune 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 mart data prune values. By default, all hourly and daily mart data older than 3 months is pruned when the process runs and all weekly and monthly mart data older than 1 year is pruned.

This process runs automatically as the last step of the ETL2 process.

The BWM.Prune\_Mart\_Log table keeps a history of the mart data removal.

### 5.2.2.1 Pruning mart data (table Prune\_Mart\_Control)

Hourly and daily data mart data is pruned from fact tables after 90 days and weekly and monthly mart data is pruned from fact tables after 1 year. 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 the date duration with the format of *yyyymmdd* (for example: *X* = 00000108 for 0000 years, 01 month, 08 days).

Change hourly data mart prune values with the following commands:

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 the following commands:

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 the following commands:

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 the following commands:

Connect to TWH\_MART

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

## 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 details on this.

### 5.3.1 Msmt data

The data extracted by the BWM\_c10\_CDW\_Process is controlled by the values contained in the TWG.Extract\_Control table in the warehouse. This table contains the sequence ID for the last data row that was extracted from the AggregateData table the last time the BWM\_c10\_CDW\_Process ran. The BWM\_c10\_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 data. Data from other Tivoli Monitoring for Transaction Performance source tables is extracted based on the AggregateData rows that correlate to those additional tables.

Note: If data is reprocessed after being inserted into the Tivoli Data Warehouse, inserting duplicate entries in the Tivoli Data Warehouse results in runtime errors because duplicate rows cannot be inserted again into the central data warehouse table.

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 IBM Tivoli Monitoring for Transaction Performance source tables whose data is extracted by central data warehouse ETL into the central data warehouse:

HOST, TRANSACTION, ARM\_USER, APPLICATION, NODE, AGGREGATEDATA, RELATIONMAP, PT, THRESHOLD, TR, MANAGEMENTPOLICY

Only the AGGREGATEDATA table uses extract control because the data pulled from the other tables is used to further define the new measurement data extracted from the AGGREGATEDATA table. Tivoli Data Warehouse, Version 1.2 uses the number and period shown before the AGGREGATEDATA names to control the data extraction from multiple source databases. See the following table for details.

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_DfTm TIMESTAMP	ExtCtl_To_DfTm TIMESTAMP	Msrc_Corr_Cd CHAR (6)
--------------------------------	--------------------------------	---------------------------------	-------------------------------	------------------------------	----------------------------	-------------------------------	-----------------------------	--------------------------

1.AGGREGATEDATA	BWM.STG_AGGREG_DATA			20	20			BWM
2.AGGREGATEDATA	BWM.STG_AGGREG_DATA			40	40			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_NAME_LONG	BWM.COMP_NAME_LONG			17	17			BWM
BWM.COMP_ATTR_LONG	BWM.COMP_ATTR_LONG			29	29			BWM

### 5.3.2 Data mart data

The data extracted by the BWM\_m05\_Mart\_Process is also controlled by the values contained in the TWG.Extract\_Control table in the warehouse. This table contains the sequence ID for the first and last data rows that were extracted the last time the BWM\_m05\_Mart\_Process ran. The BWM\_m05\_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.

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

TWG.Msmt, TWG.Comp, BWM.STG\_TX\_ND\_MET, BWM.COMP\_NAME\_LONG, BWM.COMP\_ATTR\_LONG

## 5.4 Maintenance scripts

The scripts described below 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, modify the scripts to point to that central data warehouse name.

### 5.4.1 Show\_ExtCtl\_Values.sql

The Show\_ExtCtl\_Values.sql script shows the Extract Control table values for the tables that extract control extracts for this warehouse pack.

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

```
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 runs.

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

```
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 that hold application measurement data used for reporting purposes.

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

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

```
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 the following script from the DB2 command line processor as a DB2 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\_ETL1\_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 Mart database. Trying to insert duplicate data into the BWM Mart database will cause the extract process to fail.

Use the following script only to restart the Extract Control window for the BWM\_c05\_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 databases will not cause duplicate data inserted in the BWM Mart database. Trying to insert duplicate data into the Tivoli Data Warehouse or BWM 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\_CDW\_Process and BWM\_c05\_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 the following script from the DB2 command line processor as a DB2 user that has permissions to view warehouse tables:

```
db2 -tvf Reset_ETL1_ETL2_extract_window.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 and data mart ETL 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\_Mart\_Process. This causes the CDW BWM measurement data and BWM mart data to be removed. Only use this if you know the data to be removed is no longer needed in the warehouse because the only way to replace it is by re-extracting it from the IBM Tivoli Monitoring for Transaction Performance source databases, if the data still exists there.

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

```
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 and data mart ETL 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 the following script from the DB2 command line processor as a DB2 user that has permissions to view warehouse tables:

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

## 5.5 Problem determination

If an extract process step fails, review the logs in the \$TIVOLI\_COMMON\_DIR/cdw/logs/etl directory, correct the problem, and rerun the step that failed.

See Installing and Configuring Tivoli Data Warehouse and Enabling an Application for Tivoli Data Warehouse for additional problem determination.

## 6 ETL processes

The warehouse pack has the following processes:

- BWM\_c10\_CDW\_Process
- BWM\_m05\_Mart\_Process

### 6.1 BWM\_c10\_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. This process should be run once a day before the data mart ETL is run. Measurement data is inserted into the warehouse when there are new measurements. The data that goes with a measurement such as node, host, application, etc. are only entered in the warehouse as new data when a measurement that uses those values is inserted into the warehouse.

Note: 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 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 previous data is lost.

This process has the following steps:

- **BWM\_c10\_s010\_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\_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. 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\_transform\_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.

### 6.2 BWM\_m05\_Mart\_Process

This process extracts data from the central data warehouse 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\_prepare\_stage**  
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. 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 in to fill in the dimension translation tables and hourly staging fact tables in the data mart 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 18.

Shaded columns in the following tables are translated. These columns 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
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
BWM_TRANSACTION	NULL	Transaction	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_PROBE	NULL	Monitoring Probe	2002-06-30-12.00.00.000000	9999-01-01-12.00.00.000000	BWM
BWM_HOST	NULL	Transaction Host	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)

### 7.1.3 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_DtTm TIMESTAMP	Comp_End_DtTm 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	BWM
2	IP_HOST	CDW	1		host2.ibm.com		2002-06-30-12.00.00.000000	9999-01-01-12.00.00.000000	IP HOST	SHARED
3	BWM_PROBE	CDW	1	177	QoS		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

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)
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:80/ webapp/wcs/stores /servlet/Promotion Display?promId= 10922&catalogId= - 840&storeId=1&la ngId=-1		2002-06-30- 12.00.00.000000	9999-01-01- 12.00.00.000000		BWM
8	BWM_TX_NODE	CDW	1	544	com.ibm.petstore. Session.create()_5 43	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.co m/hr/index.html_6 42	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:80/ webapp/wcs/stores /servlet/Promotion Display?promId= 10922&catalogId= - 840&storeId=1&la ngId=-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

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	J2EE_SERVER	CDW	1	673	Server2	Websphere!!5.0	2002-06-30- 12.00.00.000000	9999-01-01- 12.00.00.000000		SHARED
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.co m/*_832	host2.ibm.com@4 7!49	2002-06-30- 12.00.00.000000	9999-01-01- 12.00.00.000000		BWM
16	J2EE_DOMAIN	CDW	1	164	jdbc:db2:was40:20 03.1.4.15.51.4.539		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		BWM
18	IP_HOST	CDW	1		Stewart1.ibm.com		2002-06-30- 12.00.00.000000	9999-01-01- 12.00.00.000000	IP HOST	SHARED
19	BWM_TRANSACTION	CDW	1	349	http://www- 132.ibm.com/webap p/wcs/stores/servl et/PromotionDispl ay?promoName=5 26372&storeId=1 &catalogId=- 840&langId=- 1&dualCurrId=73		2002-06-30- 12.00.00.000000	9999-01-01- 12.00.00.000000		BWM
20	BWM_TX_NODE	CDW	1	287	http://www- 132.ibm.com/webap p/wcs/stores/servl	host2.ibm.com@5 2!61	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)
					et/PromotionDisplay?promoName=526372&storeId=1&catalogId=-840&langId=-1&dualCurrId=73_1546					
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/		2002-06-30-12.00.00.000000	9999-01-01-12.00.00.000000		BWM

\*\*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 ReInTyp)

ReInTyp_Cd CHAR (6)	ReInTyp_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

### 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	IP_HOST	RUNSON	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000
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_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

### 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
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



<b>CompReIn_ID</b> INTEGER	<b>Comp_Source_ID</b> INTEGER	<b>Comp_Target_ID</b> INTEGER	<b>ReInTyp_Cd</b> CHAR (6)	<b>CompReIn_Strt_DtTm</b> TIMESTAMP	<b>CompReIn_End_DtTm</b> TIMESTAMP	<b>MSrc_Corr_Cd</b> CHAR (6)
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	BWM
16	13	12	PCHILD	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	BWM
17	16	14	PCHILD	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	BWM
18	17	13	PCHILD	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	BWM
19	15	1	RUNSON	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	BWM
20	11	2	RUNSON	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	BWM
21	12	1	RUNSON	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	BWM

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

<b>AttrTyp_Cd CHAR (17)</b>	<b>AttrTyp_Nm * VARCHAR (120)</b>	<b>MSrc_Corr_Cd CHAR (6)</b>
WEBSITE_QUERY	Website Query	MODEL1
BWM_THRESHOLD_1	Threshold 1	BWM
BWM_THRESHOLD_2	Threshold 2	BWM
BWM_THRESHOLD_3	Threshold 3	BWM
BWM_THRESHOLD_4	Threshold 4	BWM
BWM_THRESHOLD_5	Threshold 5	BWM
BWM_THRESHOLD_6	Threshold 6	BWM
BWM_RT_CDW_ID	Warehouse Component Identifier for the Root Transaction	BWM
BWM_PT_CDW_ID	Warehouse Component Identifier for the Parent Transaction	BWM
J2EE_CELL	J2EE Cell	BWM
BWM_MGMT_POLICY	Management Policy	BWM

### 7.1.9 Attribute rule (table AttrRul)

<b>CompTyp_Cd CHAR (17)</b>	<b>AttrTyp_Cd CHAR (17)</b>	<b>AttrRul_Strt_DtTm TIMESTAMP</b>	<b>AttrRul_End_DtTm TIMESTAMP</b>	<b>AttrRul_Dom_Ind CHAR</b>	<b>AttrTyp_Multi_Val CHAR (1)</b>
IP_HOST	LAST_IP_ADDRESS	2002-07-13- 00.00.00..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)
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
BWM_TX_NODE	WEBSITE_QUERY	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	N	N
BWM_TX_NODE	BWM_THRESHOLD_1	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	N	N
BWM_TX_NODE	BWM_THRESHOLD_2	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	N	N
BWM_TX_NODE	BWM_THRESHOLD_3	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	N	N
BWM_TX_NODE	BWM_THRESHOLD_4	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	N	N
BWM_TX_NODE	BWM_THRESHOLD_5	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	N	N
BWM_TX_NODE	BWM_THRESHOLD_6	2002-06-30- 12.00.00.000000	9999-01-01 12:00:00.000000	N	N
BWM_TX_NODE	J2EE_NODE	2002-06-30-	9999-01-01	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	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	N
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)

This warehouse pack does not use the attribute domain table.

### 7.1.11 Component attribute (table CompAttr)

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)
------------------------	--------------------	----------------------	---------------------------------	--------------------------------	-------------------------------	--------------------------

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)
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/ser vlet/PromotionDispla y	BWM
10	7	WEBQUERY	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000	?promold=10922&cat alogId=-840&storeId=1&langI d=-1	BWM
11	5	BWM_THESHOLD_1	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000	1000	BWM
12	5	BWM_THESHOLD_2	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000	2000	BWM
13	7	BWM_THESHOLD_1	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000	2000	BWM
14	7	BWM_THESHOLD_2	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000	4000	BWM
15	8	BWM_MGMT_POLICY	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000	EJBPolicy	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)
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
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	WebSphere	BWM
31	10	VERSION	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000	5.0	BWM
32	9	MANUFACTURER	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000	WebSphere	BWM
33	9	VERSION	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000	5.0	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)
34	8	MANUFACTURER	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000	WebSphere	BWM
35	8	VERSION	2002-06-30-12.00.00.000000	9999-01-01 12:00:00.000000	4.6	BWM
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 CTypReln)

CTyp_Source_Cd CHAR (17)	CTyp_Target_Cd CHAR (17)	RelnTyp_Cd CHAR (6)	Reln_Strt_DtTm TIMESTAMP	Reln_End_DtTm TIMESTAMP

### 7.1.13 Component attribute type relationship (table ATypReln)

ATyp_Source_Cd CHAR (17)	ATyp_Target_Cd CHAR (17)	RelnTyp_Cd CHAR (6)	Reln_Strt_DtTm TIMESTAMP	Reln_End_DtTm 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)



<b>MGrpTyp_Cd CHAR (6)</b>	<b>MGrpTyp_Nm * VARCHAR (120)</b>
CATEG	Category
GROUP	Aggregate Types or Group Functions
* This column is translated.	

## 7.2.2 Measurement group (table MGrp)

<b>MGrp_Cd CHAR (6)</b>	<b>MGrpTyp_Cd CHAR (6)</b>	<b>MGrp_Parent_Cd CHAR (6)</b>	<b>MGrp_Nm * VARCHAR (120)</b>
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)

<b>MGrp_Cd CHAR (6)</b>	<b>MGrpTyp_Cd CHAR (6)</b>	<b>MsmtTyp_ID INTEGER</b>
AVG_E	GROUP	1
AVG_E	GROUP	2
AVG_E	GROUP	3
AVG_E	GROUP	4

<b>MGrp_Cd CHAR (6)</b>	<b>MGrpTyp_Cd CHAR (6)</b>	<b>MsmfTyp_ID INTEGER</b>
AVG_E	GROUP	6
AVG_E	GROUP	7
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

### 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	ReInTyp_Cd CHAR (6)	MReIn_Strt_DtTm TIMESTAMP	MReIn_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)
Tivoli	NULL	Tivoli Application
BWM	Tivoli	IBM Tivoli Monitoring for Transaction Performance v 5.2.0

### 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 MsmTyp)

MsmTyp_ID INTEGER	MUnit_Cd CHAR (6)	MSrc_Cd CHAR (6)	MsmTyp_Nm * VARCHAR (120)	MsmTyp_Ds * VARCHAR (254)
1	MSec	MODEL1	Response Time	Response Time
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	Percentage of unsuccessful synthetic transactions

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

### 7.2.11 Component measurement rule (table MsmtRul)

<b>CompTyp_Cd</b> <b>CHAR (17)</b>	<b>MsmtTyp_ID</b> <b>INTEGER</b>
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

## 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)
4	9	1	H	2002-06-30	13:00:00	300	1078	502		248	8		BWM
5	9	2	H	2002-06-30	13:00:00	358	6052	3054		248	8		BWM
6	9	5	H	2002-06-30	13:00:00				20	256	0		BWM
7	8	1	H	2002-06-30	13:00:00	300	1078	502		253	3		BWM
8	8	2	H	2002-06-30	13:00:00	358	6052	3054		253	3		BWM
9	8	5	H	2002-06-30	13:00:00				15	255	1		BWM
10	15	1	H	2002-06-30	13:00:00	100	200	150		356	0		BWM
11	15	2	H	2002-06-30	13:00:00	406	1000	675		356	0		BWM

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

### 7.2.13 Threshold measurement objective (table Mobj)

<b>Mobj_ID</b> INTEGER	<b>MsmtTyp_ID</b> INTEGER	<b>CompTyp_Cd</b> CHAR (17)	<b>Centr_Cd</b> CHAR (6)	<b>Cust_ID</b> INTEGER	<b>Attrdom_ID</b> INTEGER	<b>Msrc_Cd</b> CHAR (6)	<b>Mobj_Strt_DtTm</b> TIMESTAMP	<b>Mobj_End_DtTm</b> TIMESTAMP

### 7.2.14 Threshold measurement objective range (table MobjRng)

Mobjrng_ID INTEGER	Mobj_ID INTEGER	Sev_Cd CHAR	Mobjrng_Min_Val VARCHAR (254)	Mobjrng_Max_Val VARCHAR (254)	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_CD 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\_NAME\_LONG)

The component long table is used to store component names that are longer than the 254 characters allowed in the component table (Comp).

Comp_ID INTEGER	Comp_Nm VARCHAR (4000)	Comp_DS VARCHAR (254)	Comptyp_cd VARCHAR(17)
1972	http://stewart.ibm.com:9090/admin/com.ibm.ws.console.resources.forwardCmd.do?forwardName=MQQueue.content.main&sfname=factories&resourceUri=resources.xml&parentRefId=builtin_mqprovider&contextId=cells:	URI	BWM_TRANSACTION



Comp_ID INTEGER	Comp_Nm VARCHAR (4000)	Comp_DS VARCHAR (254)	Comptyp_cd VARCHAR(17)
	stewart:nodes:stewart&perspective=tab.configuration		

### 7.3.2 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=MQQueue.content.main&sfname=factories&resourceUri=resources.xml&parentRefId=builtin_mqprovider&contextId=cells:stewart:nodes:stewart&perspective=tab.configuration	WEBSITE_QUERY

## 7.4 Exception tables

This warehouse pack does not currently generate exception tables.

## 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 perform the following actions:

- Extract data from the IBM Tivoli Monitoring for Transaction Performance source database and store it into the central data warehouse during the central data warehouse ETL.
- Extract data from the central data warehouse and store it into the data mart tables during the data mart ETL.

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 and the data mart ETL 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 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 process. The rest are the table names are the source tables extracted from during the data mart ETL 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.STG_AGGREG_DATA	-1	-1
2.AGGREGATEDATA	BWM.STG_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, data marts, and reports 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 18.

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_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_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_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_TX
	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

### **8.3 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

## 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)
0	Not Used	Number of Times a transaction was executed	Number of Executions	QTY	N	N	N	Y	IBM Tivoli Monitoring For Transaction Performance v 5.2
1	Not Used	The backend service response time	Service Time	Sec	Y	Y	Y	N	IBM Tivoli Monitoring For Transaction Performance v 5.2
2	Not Used	The number of transaction	Number Threshold	QTY	N	N	N	Y	IBM Tivoli Monitoring For Transaction

<b>Metric_ID INTEGER</b>	<b>Met_Category * VARCHAR (254)</b>	<b>Met_Desc * VARCHAR (254)</b>	<b>Met_Name * VARCHAR (254)</b>	<b>Met_Units * VARCHAR (254)</b>	<b>Min_Exists CHAR (1)</b>	<b>Max_Exists CHAR (1)</b>	<b>Avg_Exists CHAR (1)</b>	<b>Total_Exists CHAR (1)</b>	<b>Msrc_Nm * VARCHAR (254)</b>
		thresholds exceeded	Exceeded						Performance v 5.2
3	Not Used	The page render response time	Page Render Time	Sec	Y	Y	Y	N	IBM Tivoli Monitoring For Transaction Performance v 5.2
4	Not Used	The percentage of synthetic transactions that failed	Unsuccessful Transactions	PRC	Y	Y	Y	N	IBM Tivoli Monitoring For Transaction Performance v 5.2
5	Not Used	The percentage of synthetic transactions that were successful	Successful Transactions	PRC	Y	Y	Y	N	IBM Tivoli Monitoring For Transaction Performance v 5.2
6	Not Used	The round trip response time	Round Trip Time	Sec	Y	Y	Y	N	IBM Tivoli Monitoring For Transaction Performance v 5.2
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
* This column is translated.									

## 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\_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

### 8.5.3 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

#### **8.5.4 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

### ***8.6 Mart translation tables***

The following tables are utilized by the data mart ETL to move data from the central data warehouse to the data marts using extract control to only extract the newly added data since the last data mart ETL ran. 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\_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

#### **8.6.4 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

#### **8.6.5 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







## 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  
2Z4A/101  
11400 Burnet Road  
Austin, TX 78758 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

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.

SC32-1388-01