IBM Tivoli Monitoring 5.1 Warehouse Enablement Pack: Implementation Guide

Date of Last Update: **4 March, 2004**

About this Document

The Tivoli Enterprise Data Warehouse infrastructure enables a set of extract, transform, and load (ETL) utilities to extract and move data from Tivoli application data stores to a central data warehouse database. Database administrators and programmers require the type of information provided in this document to install the ETL tools and understand what data is being populated in the central data warehouse during what is referred to as the *central data warehouse ETL*.

NOTE: This updated WEP replaces the original AMW WEP and is required to overcome the installation problems on a TDW 1.2 environment. You should use it, instead of any other AMW WEP, only if you are installing on a TDW 1.2 environment. Refer to the TDW 1.2 documentation for further details on how to install a new WEP.

Overview of Tivoli Enterprise Data Warehouse

Tivoli Enterprise 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
Report interfaces

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

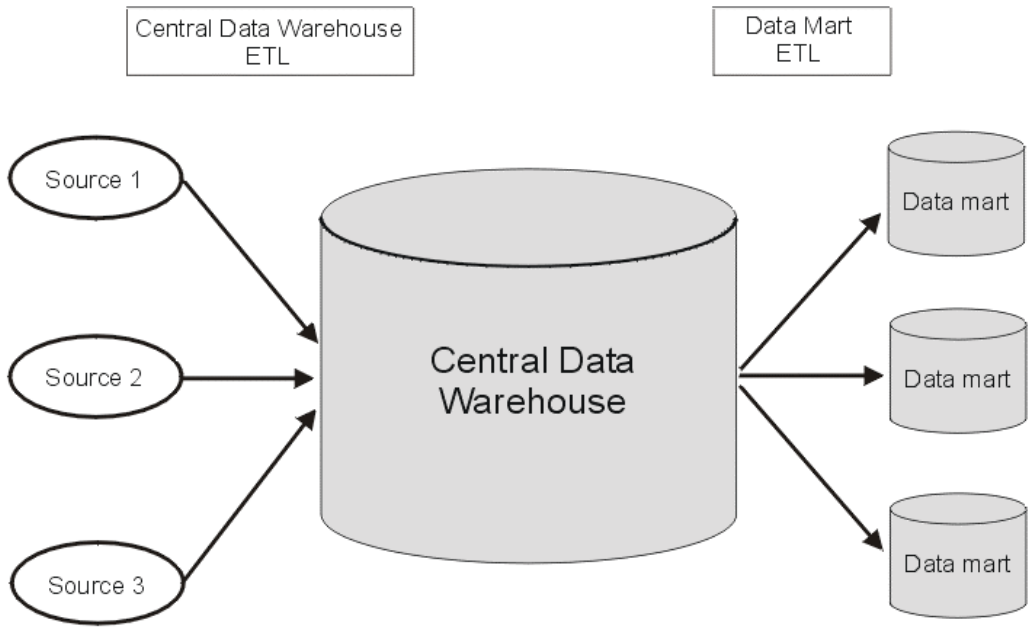

Figure 1. Tivoli Enterprise Data Warehouse overview

The *central data warehouse* uses a generic schema. As new components or new applications are added, more data is added to the database; however, no new tables or columns are added in the schema.

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

Design and Data Flow

The data that will be stored in the central data warehouse is retrieved by a IBM Tivoli Monitoring schema defined in one of the following databases:

- Oracle
- Sybase
- DB2
- Informix
- MS-SQL

This data is gathered by monitors that are contained in a specific IBM Tivoli Monitoring profile, SPR_NtProfile (to be used on the NT and Windows 2000 platform), SPR_UnixProfile (to be used on the Unix platform). These monitors are run on the target box and the hourly aggregated data are stored in files on the box itself that is managed by IBM Tivoli Monitoring.
Nightly tasks are run to maintain backup files containing data and to store these hourly aggregated data in the database, in the table DM_METRICS.
The process is the following:

1) Job DMAE_ReadDataInDB creates log files reading data contained in the endpoint database and stored there by IBM Tivoli Monitoring engine all over the day. Besides it backups these files.

This last step is needed to allow the possibility of aggregating data for a certain time in the future to avoid the loss of data. (IBM Tivoli Monitoring engine stores data in its database in a cyclic way.)

2) Job DMAE_NewDataAggregation reads the data from the files created by the job above and aggregates them in a single file containing all together the following values:

minimum value for each hour
maximum value for each hour
average value for each hour
minimum value in the day
maximum value in the day
average value in the day
minimum value between all the hours
maximum value between all the hours

These values are stored in a file on TMR Server.

3) Job DMAE_RollupIntoDB reads the file produced by the previous job and inserts them in the table DM_METRICS via RIM object.

Once this data is stored in the central data warehouse, it is possible to run the central data warehouse ETL process, which is based on the situation explained above.

The central data warehouse ETL process consists of the following:

1) Data stored in DM_METRICS is read, and stored in some temporary tables defined in the same DB2 database in which the central data warehouse resides, but in an area specific for IBM Tivoli Monitoring.

2) Some of the tables in the central data warehouse generic data model are filled in with IBM Tivoli Monitoring data.

At this point all data produced by IBM Tivoli Monitoring is available in the central data warehouse and every application that needs them can start from here, running a data mart ETL process that reads data from the central data warehouse and stores it in the data mart data model specific for the consumer application.

The IBM Tivoli Monitoring central data warehouse ETL process will fill in all the fields related to the end_date. The default value is '9999-01-01 00:00:00.0'. It is not possible to update this value with an actual date, representing the fact that the object is no longer enabled. IBM Tivoli Monitoring does not have knowledge regarding this event.

1.1 How to implement incremental extract

The incremental extract for IBM Tivoli Monitoring will be implemented in the following way.
Table TWH_DM_METRICS is created with all the primary key columns from DM_METRICS plus TWH_INSERT_SEQ and TWH_GMTOFFSET_MIN fields. This table is created, in the same RDBMS in which the table DM_METRICS is created, by the application of the prerequisite patch.
In aggreg.pl, the number of minutes offset from GMT is set and it is pad to four chars ie '-720' .... '+720' – this is called gmt_offset_min.
gmt_offset_min is concatenated to the Profile_Collection string to make '-720Name_of_profile_collection' - assuming only 60 of the 64 characters are used.
The trigger on insert into DM_METRICS is changed so that:
An insert is made into TWH_DM_METRICS with all the primary key columns from DM_METRICS
The first four characters from Profile_Collection are copied into TWH_DM_METRICS.TWH_GMT_OFFSET_MIN
Characters 5 to 64 in Profile_Collection are used to update Profile_Collection (ie shift left 4) in DM_METRICS
TWH_INSERT_SEQ is set using the Tivoli Enterprise Data Warehouse ascending sequence function (different per RDBMS).
When the incremental extract is run the following actions are done:
Table temp_extract_control is created and updated, retrieving the value for the field extctl_from_rawseq from Extract_Control table in Tivoli Enterprise Data Warehouse and the value for the field extctl_to_rawseq from the TWH_DM_METRICS table. (It is actually the maximum value contained in the field TWH_INSERT_SEQ.)
Table STAGE_DM_METRICS is created in IBM Tivoli Monitoring RDBMS and its values are retrieved from the tables DM_METRICS, TWH_DM_METRICS and temp_extract_control. This is the step in which data is read in the IBM Tivoli Monitoring source schema and stored in the Tivoli Enterprise Data Warehouse schema.
Table temp_to_rawseq is created in the IBM Tivoli Monitoring RDBMS and its values are retrieved from the table temp_extract_control. It contains the last sequence number extracted.
Table extract_control in Tivoli Enterprise Data Warehouse is updated with the value retrieved from temp_to_rawseq and the table Extract_Log is also updated with the values retrieved from extract_control table.

The extract window is copied over to the source database into a temp table then the extract is done based on a join between a table in the source database and this temporary table.
This takes away the requirement to store the Extract_Control and Extract_Log tables as permanent tables in the source database.

1.2 Source Database Information

The source database for the ETL has to be the database in which IBM Tivoli Monitoring is storing data using the TDS component. This means that the source database is the database for which the RIM object spr_rim is configured. An available RDBMS in which IBM Tivoli Monitoring can store data can be Oracle, MS SQL, Sybase[1], DB2 and Informix.

[1] In order to allow DDL statements in transaction you have to run the following commands as a user with the 'sa_role' granted: *use master go sp_dboption dm_db, 'ddl in tran', true go*.

1.3 Install Information

The installation instructions related to IBM Tivoli Monitoring are included in IBM Tivoli Monitoring manuals and the patch instructions are contained in the README that is provided in the patch itself. See *Installing and Configuring Tivoli Enterprise Data Warehouse* for instructions on how to install Tivoli Enterprise Data Warehouse.

The ETL process used to perform the in-flow of data from the specific IBM Tivoli Monitoring repository to the Tivoli Enterprise Data Warehouse central data warehouse tables requires a system data source to be defined on the machine that hosts the RDBMS used by Tivoli Enterprise Data Warehouse. This data source must be named AMW_RIM and must use one of the Merant ODBC drivers provided by DB2, according to the specific IBM Tivoli Monitoring repository (Oracle 8.0.5, Microsoft SQLServer 7.0, Sybase 11.9.2, Informix 7.3, DB2 6.1).

The enablement for Tivoli Enterprise Data Warehouse also requires the application of a patch on the IBM Tivoli Monitoring repository, but if you have data collected by IBM Tivoli Monitoring before the application of this patch, you can run the initDMData process; after defining the maximum number of days to be considered during the data loading with an update to the Prune_Msmt_Control table; for example, if you are interested in the latest 2 months of IBM Tivoli Monitoring Data, the SQL statement is:

```
UPDATE TWG.Prune_Msmt_Control
        SET PMSMTC_AGE_IN_DAYS = 200
      WHERE TMSUM_CD = 'H'
        AND MSRC_CD  = 'AMW'
;
```

As the PMsmtC_Age_In_Days column is a date duration whose format is yyyymmdd.

From within the Data Warehouse Center, you can use the process AMW_c05_initDMData_Process to perform the initial loading of IBM Tivoli Monitoring data. When the patches are applied, you can perform the ETL process by running the AMW_c10_loadDMData_Process.

Before running the process above, you must be sure that the Warehouse Sources and the Warehouse Targets specific to the AMW application are correctly defined (username, password, hostname, system data source). Note also that, in order to run the provided sample reports, you must update the sample data mart properties (SAMPLE1) with your own connection data (username, password, JDBC URL).

1.4 Initialization Scripts

In order to fill in the data warehouse database, all the tables should have been created and initialized if necessary.

Tivoli Enterprise Data Warehouse will provide DDL/DML scripts to:

- (DDL) create the tables in the generic schema
- (DML) populate the generic schema with some static data such as the measurement unit and measurement category.

All the DDL and DML scripts will be called by the Tivoli Enterprise Data Warehouse installation procedures.

IBM Tivoli Monitoring will provide DDL/DML scripts:

(DDL) to create the temporary tables, if they are needed
(DML) to populate the generic schema with static data such as:
Data that describes the type of components for IBM Tivoli Monitoring into the table CompTyp
Data that describes the relationship rules for IBM Tivoli Monitoring Components into the table RelnRul
Data that describes the type of monitors for IBM Tivoli Monitoring into the table MsmtTyp.
Data that describes the Time summary into the TmSum.
Data that describes the Measurement Rules into MsmtRul.
Data that describes the measurement units into the MUnit table.
Data that describes the measurement unit categories into the MUnitCat table.
Data that describes the measurement groups into the MGrp table.
Data that describes the measurement group types into the MGrpTyp table.
Data that describes the measurement group members into the MGrpMbr table.

Each script has its corresponding remove script.

1.5 Migrating from DM 4.1 TEDW Enablement

If you have previously installed and used the DM 4.1 TEDW Enablement and now you want to migrate to the 5.1 version, you must uninstall the DM 4.1 TEDW Enablement before installing the IBM – Tivoli Monitoring 5.1 TEDW Enablement pack.

In order to cleanly uninstall the DM 4.1 TEDW Enablement pack, preventing data losses, you must use the twh_app_deinstall.sh script from the install\bin directory under the TWH installation root, after editing the twh_app_deinstall.cfg in order to define the application to be removed; i.e. you must set:

```
APPLICATION_TO_DELETE=AMW
```

in the twh_app_deinstall.cfg file and then run the command:

```
./twh_app_deinstall.sh -config ./twh_app_deinstall.cfg
```

from within a bash shell opened on the <TWH_INSTALL_ROOT>\install\bin directory.

After successfully uninstalling the DM 4.1 TEDW Enablement pack you can proceed with the installation of the IBM – Tivoli Monitoring 5.1 TEDW Enablement pack using the installation GUI as usual.

The new Enablement pack can still work with data collected by the previous one, but it adds some sample reports about CPU and Network performance that you can run and display using the Web-Based IBM Console (see section 1.3 for installation informations).

Special Installation Instructions

After installing the IBM Tivoli Monitoring 5.1 base package and all the collections needed in your environment, install the Tivoli Distributed Monitoring TDS Configuration package. Note that this package name contains the term "TDS" because it was designed to gather data for Tivoli Decision Support for Server Performance Prediction (also known as the "SPP guide"). The Tivoli Distributed Monitoring TDS Configuration package is also used to gather data for use with Tivoli Enterprise Data Warehouse; however, neither the SPP guide nor any additional Tivoli Decision Support prerequisites are required.

It is strongly recommended that you install the latest level of patches available for IBM Tivoli Monitoring 5.1 related to the TDS component. This will allow you to work with the most current version of it that gathers data feeding the Tivoli Enterprise Data Warehouse.

You can find detailed information related to these patches in their specific README files as well as any actions that should occur after installing the patches.

Generic Schema Implementation

Refer to the Tivoli Enterprise Data Warehouse central data warehouse generic schema before reading this section. You will find the definitions and the relation of all the tables in this document. While the content of the tables changes from application to application, most applications will have data in all of the tables shown in this section. All the columns may not be used. These do not contain any information

Note: Central data warehouse columns marked with an asterisk (*) indicate values that source applications are responsible for translating and delivering corresponding Java resource bundles.

1.6 Component Configuration

1.6.1 Component Type (table CompTyp)

CompTyp_Cd CHAR(17)	CompTyp_Nm* VARCHAR(120)	CompTyp_Parent_Cd CHAR(17)	CompTyp_Strt_DtTm TIMESTAMP	CompTyp_End_DtTm TIMESTAMP
IP_HOST	IP Host	NULL	1/1/1970 12:00:00 PM	1/1/9999 12:00:00 PM
IP_INTERFACE	IP Interface	NULL	1/1/1970 12:00:00 PM	1/1/9999 12:00:00 PM
APP_HOST	Application Specific Hosts	NULL	1/1/1970 12:00:00 PM	1/1/9999 12:00:00 PM
TME_ENDPOINT	TME Endpoint	APP_HOST	1/1/1970 12:00:00 PM	1/1/9999 12:00:00 PM
AMW_HOST	AMW Host	APP_HOST	1/1/1970 12:00:00 PM	1/1/9999 12:00:00 PM
WIN_AMW_HOST	Windows AMW Host	AMW_HOST	1/1/1970 12:00:00 PM	1/1/9999 12:00:00 PM
UNIX_AMW_HOST	Unix AMW Host	AMW_HOST	1/1/1970 12:00:00 PM	1/1/9999 12:00:00 PM
WIN_IP_HOST	Windows IP Host	IP_HOST	1/1/1970 12:00:00 PM	1/1/9999 12:00:00 PM
UNIX_IP_HOST	Unix IP Host	IP_HOST	1/1/1970 12:00:00 PM	1/1/9999 12:00:00 PM

NOTE: Windows NT, Windows 2000 and Windows XP hosts are classified as Windows AMW/IP Host.

1.6.2 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)
0	AMW_HOST	CDW	1	0	Ntamwhost	AMW	2/2/2002 11:00:00 AM	1/1/9999 12:00:00 PM	NULL
1	WIN_AMW_HOST	CDW	1	0	WIN_AMW_DATA	AMW	2/2/2002 11:00:00 AM	1/1/9999 12:00:00 PM	AMW_DATA
2	IP_HOST	CDW	1	0	ntiphost.tivoli.com	AMW	2/2/2002 11:00:00 AM	1/1/9999 12:00:00 PM	NULL
3	WIN_IP_HOST	CDW	1	0	WIN_IP_DATA	AMW	2/2/2002 11:00:00 AM	1/1/9999 12:00:00 PM	AMW_DATA
4	AMW_HOST	CDW	1	0	unixamwhost	AMW	2/2/2002 11:00:00 AM	1/1/9999 12:00:00 PM	NULL
5	UNIX_AMW_HOST	CDW	1	0	UNIX_AMW_DATA	AMW	2/2/2002 11:00:00 AM	1/1/9999 12:00:00 PM	AMW_DATA
6	IP_HOST	CDW	1	0	unixiphost.tivoli.com	AMW	2/2/2002 11:00:00 AM	1/1/9999 12:00:00 PM	NULL
7	UNIX_IP_HOST	CDW	1	0	UNIX_IP_DATA	AMW	2/2/2002 11:00:00 AM	1/1/9999 12:00:00 PM	AMW_DATA

* Source applications must translate values in this column and deliver corresponding Java resource bundles.

1.6.3 Component Relationship Type (table RelnTyp)

Defines the types of relationships that may be captured about two independent components. For example: a WIN_AMW_HOST is a specialization of a more general AMW_HOST so they are in a Parent-Child relationship.

RelnTyp_Cd CHAR(6)	RelnTyp_Nm* VARCHAR(120)
PCHILD	Parent Child Relation

1.6.4 Component Relationship Rule (table RelnRul)

Defines which types of components may participate in which types of relationships. Rules are enabled/disabled using the start and end dates/times, but because IBM Tivoli Monitoring relationships between component types are permanent, the start to end period is chosen conveniently large.

CompTyp_Source_Cd CHAR(17)	CompTyp_Target_Cd CHAR(17)	RelnTyp_Cd CHAR(6)	RelnRul_Strt_DtTm TIMESTAMP	RelnRul_End_DtTm TIMESTAMP
AMW_HOST	UNIX_AMW_HOST	PCHILD	1/1/1970 12:00:00 PM	1/1/9999 12:00:00 PM
IP_HOST	UNIX_IP_HOST	PCHILD	1/1/1970 12:00:00 PM	1/1/9999 12:00:00 PM
AMW_HOST	WIN_AMW_HOST	PCHILD	1/1/1970 12:00:00 PM	1/1/9999 12:00:00 PM
IP_HOST	WIN_IP_HOST	PCHILD	1/1/1970 12:00:00 PM	1/1/9999 12:00:00 PM
APP_HOST	AMW_HOST	PCHILD	1/1/1970 12:00:00 PM	1/1/9999 12:00:00 PM

1.6.5 Component Relationship (table CompReln)

Defines a connection, logical or physical, between two separate components. For IBM Tivoli Monitoring, it defines the relationship between a specific host and the OS running on it (WIN or UNIX).
The following values are used to represent parent-child relationships:

RelnTyp_Cd = 'PCHILD'
Comp_Target_ID = ID of the Parent Component
Comp_Source_ID = ID of the Child Component

CompReln_ID INTEGER	Comp_Source_ID INTEGER	Comp_Target_ID INTEGER	RelnTyp_Cd CHAR(6)	CompReln_Strt_DtTm TIMESTAMP	CompReln_End_DtTm TIMESTAMP
1	0	1	PCHILD	1/1/1970 12:00:00 PM	1/1/9999 12:00:00 PM
2	2	3	PCHILD	1/1/1970 12:00:00 PM	1/1/9999 12:00:00 PM
3	4	5	PCHILD	1/1/1970 12:00:00 PM	1/1/9999 12:00:00 PM
4	6	7	PCHILD	1/1/1970 12:00:00 PM	1/1/9999 12:00:00 PM

1.6.6 Attribute Type (table AttrTyp)

THIS IS NOT USED BY IBM Tivoli Monitoring.

AttrTyp_Cd CHAR(17)	AttrTyp_Nm* VARCHAR(120)

1.6.7 Attribute Rule (table AttrRul)

THIS IS NOT USED BY IBM Tivoli Monitoring.

CompTyp_Cd CHAR(17)	AttrTyp_Cd CHAR(17)	AttrRul_Strt_DtTm TIMESTAMP	AttrRul_End_DtTm TIMESTAMP	AttrRul_Dom_Ind CHAR

1.6.8 Attribute Domain (table AttrDom)

THIS IS NOT USED BY IBM Tivoli Monitoring.

AttrDom_Id INTEGER	CompTyp_Cd CHAR(17)	AttrTyp_Cd CHAR(17)	AttrDom_ Strt_DtTm TIMESTAMP	AttrDom_ End_DtTm TIMESTAMP	AttrDom_Val VARCHAR(254)	AttrDom_Ds VARCHAR(254)

1.6.9 Component Attribute (table CompAttr)

THIS IS NOT USED BY IBM Tivoli Monitoring.

CompAttr_ID INTEGER	Comp_ID INTEGER	AttrTyp_Cd CHAR(17)	CompAttr_Strt_DtTm TIMESTAMP	CompAttr_End_DtTm TIMESTAMP	CompAttr_Val VARCHAR(254)

1.7 Component Measurement

1.7.1 Measurement Group Type (table MGrpTyp)

MgrpTyp_Cd CHAR(6)	MGrpTyp_Nm* VARCHAR(120)
DM2	CPU/MEM/NW/IO
GROUP	Aggregate Types or Group Functions

* Source applications must translate values in this column and deliver corresponding Java resource bundles
* Source applications must translate values in this column and deliver corresponding Java resource bundles

1.7.2 Measurement Group (table MGrp)

MGrp_Cd CHAR(6)	MGrpTyp_Cd CHAR(6)	MGrp_Parent_Cd CHAR(6)	MGrp_Nm VARCHAR(120)
MIN_E	GROUP	NULL	Minimum value exists
MAX_E	GROUP	NULL	Maximum value exists
AVG_E	GROUP	NULL	Average value exists
TOT_E	GROUP	NULL	Total value exists
CPU_E	DM2	NULL	CPU
NIC_E	DM2	NULL	Network interface
FS_E	DM2	NULL	File system
MEM_E	DM2	NULL	Memory
HD_E	DM2	NULL	Disk

1.7.3 Measurement Group Member (table MGrpMbr)

Defines a type of measurement within a particular grouping (Measurement Group). A Measurement Type may be a member of several Measurement Groups.

MGrp_Cd CHAR(6)	MGrpTyp_Cd CHAR(6)	MsmtTyp_ID INTEGER
CPU_E	DM2	1
CPU_E	DM2	2
CPU_E	DM2	3
CPU_E	DM2	4
CPU_E	DM2	5
CPU_E	DM2	6
CPU_E	DM2	18
CPU_E	DM2	19
CPU_E	DM2	20
CPU_E	DM2	31
CPU_E	DM2	32
FS_E	DM2	7
FS_E	DM2	8
FS_E	DM2	30
HD_E	DM2	28
HD_E	DM2	29
MEM_E	DM2	9
MEM_E	DM2	10
MEM_E	DM2	11
MEM_E	DM2	12
MEM_E	DM2	21
MEM_E	DM2	22
MEM_E	DM2	23
NIC_E	DM2	13
NIC_E	DM2	14
NIC_E	DM2	15
NIC_E	DM2	16
NIC_E	DM2	17
NIC_E	DM2	24
NIC_E	DM2	25
NIC_E	DM2	26
NIC_E	DM2	27
AVG_E	GROUP	1
AVG_E	GROUP	2
AVG_E	GROUP	3
AVG_E	GROUP	4
AVG_E	GROUP	5

AVG_E	GROUP	6
AVG_E	GROUP	7
AVG_E	GROUP	8
AVG_E	GROUP	9
AVG_E	GROUP	10
AVG_E	GROUP	11
AVG_E	GROUP	12
AVG_E	GROUP	13
AVG_E	GROUP	14
AVG_E	GROUP	15
AVG_E	GROUP	16
AVG_E	GROUP	17
AVG_E	GROUP	18
AVG_E	GROUP	19
AVG_E	GROUP	20
AVG_E	GROUP	21
AVG_E	GROUP	22
AVG_E	GROUP	23
AVG_E	GROUP	24
AVG_E	GROUP	25
AVG_E	GROUP	26
AVG_E	GROUP	27
AVG_E	GROUP	28
AVG_E	GROUP	29
AVG_E	GROUP	30
AVG_E	GROUP	31
AVG_E	GROUP	32
MAX_E	GROUP	1
MAX_E	GROUP	2
MAX_E	GROUP	3
MAX_E	GROUP	4
MAX_E	GROUP	5
MAX_E	GROUP	6
MAX_E	GROUP	7
MAX_E	GROUP	8
MAX_E	GROUP	9
MAX_E	GROUP	10
MAX_E	GROUP	11
MAX_E	GROUP	12
MAX_E	GROUP	13
MAX_E	GROUP	14
MAX_E	GROUP	15
MAX_E	GROUP	16
MAX_E	GROUP	17
MAX_E	GROUP	18
MAX_E	GROUP	19
MAX_E	GROUP	20
MAX_E	GROUP	21
MAX_E	GROUP	22
MAX_E	GROUP	23
MAX_E	GROUP	24
MAX_E	GROUP	25
MAX_E	GROUP	26
MAX_E	GROUP	27
MAX_E	GROUP	28
MAX_E	GROUP	29
MAX_E	GROUP	30
MAX_E	GROUP	31
MAX_E	GROUP	32
MIN_E	GROUP	1
MIN_E	GROUP	2
MIN_E	GROUP	3
MIN_E	GROUP	4
MIN_E	GROUP	5
MIN_E	GROUP	6
MIN_E	GROUP	7
MIN_E	GROUP	8

MIN_E	GROUP	9
MIN_E	GROUP	10
MIN_E	GROUP	11
MIN_E	GROUP	12
MIN_E	GROUP	13
MIN_E	GROUP	14
MIN_E	GROUP	15
MIN_E	GROUP	16
MIN_E	GROUP	17
MIN_E	GROUP	18
MIN_E	GROUP	19
MIN_E	GROUP	20
MIN_E	GROUP	21
MIN_E	GROUP	22
MIN_E	GROUP	23
MIN_E	GROUP	24
MIN_E	GROUP	25
MIN_E	GROUP	26
MIN_E	GROUP	27
MIN_E	GROUP	28
MIN_E	GROUP	29
MIN_E	GROUP	30
MIN_E	GROUP	31
MIN_E	GROUP	32

1.7.4 Measurement Unit Category (table MUnitCat)

A high level classification of measurements as follows:

MUnitCat_Cd CHAR(6)	MUnitCat_Nm* VARCHAR(120)
QTY	Quantity
TM	Time Duration
PRC	Percentage
RT	Rate

1.7.5 Measurement Unit (table MUnit)

MUnit_Cd CHAR(6)	MUnitCat_Cd CHAR(6)	MUnit_Nm* VARCHAR(120)
MB	QTY	Megabytes
Qps	RT	Quantity per second
Qpm	RT	Quantity per minute
QTY	QTY	Quantity
KBps	RT	Kilobytes per second
Bps	RT	Bytes per second
PRC	PRC	Percentage

1.7.6 Time Summary (table TmSum)

The period over which a measurement may be summarized.

TmSum_Cd CHAR	TmSum_Nm* VARCHAR(120)

* Source applications must translate values in this column and deliver corresponding Java resource bundles

H	Hourly
D	Daily

1.7.7 Measurement Source (table MSrc)

MSrc_Cd CHAR(6)	MSrc_Parent_Cd CHAR(6)	MSrc_Nm* VARCHAR(120)
Tivoli	NULL	Tivoli Applications
AMW	Tivoli	IBM Tivoli Monitoring 5.1

1.7.8 Measurement Type (table MsmtType)

Refer to 1.7.5 Measurement Unit for the meaning of the MUnit_Cd column.

MsmtTyp_ID INTEGER	MSrc_Cd CHAR (6)	MUnit_Cd CHAR(6)	MsmtTyp_Nm* VARCHAR(120)	MsmtTyp_Ds* VARCHAR(254)
1	AMW	PRC	idleTime	CPU percent busy
2	AMW	PRC	userTime	CPU percent user time
3	AMW	PRC	sysTime	CPU percent system time
4	AMW	QTY	loadAvg1	CPU run queue length
5	AMW	QTY	numberWaitProcesses	Number of processes totally waiting
6	AMW	QTY	numOfProcesses	Number of running processes
7	AMW	KB	availKBytes	Disk Space available
8	AMW	PRC	prcInodeUsed	Inode used in percent per file system
9	AMW	Qps	pageInsRate	Memory page-in rate
10	AMW	Qps	pageOutsRate	Memory page-out rate
11	AMW	MB	availSwapSpace	Swap space available
12	AMW	PRC	pctusedVirtualStorage	Used virtual storage in percent
13	AMW	Qps	deltaInPackets	Network packet input rate
14	AMW	Qps	deltaInPacketsErr	Network packet input error rate
15	AMW	Qps	deltaOutPackets	Network packet output rate
16	AMW	Qps	deltaOutPacketsErr	Network packet output error rate
17	AMW	Qps	deltaCollisions	Network packet collision rate
18	AMW	QTY	NumberOfProcesses	Number of running processes
19	AMW	PRC	PrcTotUserTime	CPU percent user time
20	AMW	QTY	ProcessorQueueLength	CPU run queue length
21	AMW	Qps	PageInputSec	Memory page-in rate
22	AMW	Qps	PageOutputSec	Memory page-out rate
23	AMW	QTY	Avail	Memory available
24	AMW	Qps	PacketsReceivedSec	Network packet input rate
25	AMW	Qps	PacketsReceivedErrors	Network packet input error rate
26	AMW	Qps	PacketsSentSec	Network packet output rate
27	AMW	Qps	PacketsOutboundErrors	Network packet output error rate
28	AMW	Bps	DiskBytesSec	Disk input / output rate
29	AMW	Qps	DiskXfersSec	Disk transfer rate
30	AMW	MB	FreeMB	Disk space available
31	AMW	PRC	PrcTotCpuTime	CPU percent busy
32	AMW	PRC	PrcTotPrivTime	CPU percent system time

1.7.9 Component Measurement Rule (table MsmtRul)

Defines which types of measurements are appropriate for which types of components.

CompTyp_Cd CHAR(17)	MsmtTyp_ID INTEGER

UNIX_AMW_HOST	1
UNIX_AMW_HOST	2
UNIX_AMW_HOST	3
UNIX_AMW_HOST	4
UNIX_AMW_HOST	5
UNIX_AMW_HOST	6
UNIX_AMW_HOST	7
UNIX_AMW_HOST	8
UNIX_AMW_HOST	9
UNIX_AMW_HOST	10
UNIX_AMW_HOST	11
UNIX_AMW_HOST	12
UNIX_AMW_HOST	13
UNIX_AMW_HOST	14
UNIX_AMW_HOST	15
UNIX_AMW_HOST	16
UNIX_AMW_HOST	17
UNIX_IP_HOST	1
UNIX_IP_HOST	2
UNIX_IP_HOST	3
UNIX_IP_HOST	4
UNIX_IP_HOST	5
UNIX_IP_HOST	6
UNIX_IP_HOST	7
UNIX_IP_HOST	8
UNIX_IP_HOST	9
UNIX_IP_HOST	10
UNIX_IP_HOST	11
UNIX_IP_HOST	12
UNIX_IP_HOST	13
UNIX_IP_HOST	14
UNIX_IP_HOST	15
UNIX_IP_HOST	16
UNIX_IP_HOST	17
WIN_AMW_HOST	18
WIN_AMW_HOST	19
WIN_AMW_HOST	20
WIN_AMW_HOST	21
WIN_AMW_HOST	22
WIN_AMW_HOST	23
WIN_AMW_HOST	24
WIN_AMW_HOST	25
WIN_AMW_HOST	26
WIN_AMW_HOST	27
WIN_AMW_HOST	28
WIN_AMW_HOST	29
WIN_AMW_HOST	30
WIN_AMW_HOST	31
WIN_AMW_HOST	32
WIN_IP_HOST	18
WIN_IP_HOST	19
WIN_IP_HOST	20
WIN_IP_HOST	21
WIN_IP_HOST	22
WIN_IP_HOST	23
WIN_IP_HOST	24
WIN_IP_HOST	25
WIN_IP_HOST	26
WIN_IP_HOST	27
WIN_IP_HOST	28
WIN_IP_HOST	29
WIN_IP_HOST	30
WIN_IP_HOST	31
WIN_IP_HOST	32

1.7.10 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_Sampl_Cnt INTEGER	Msmt_Err_Cnt INTEGER
1	1	1	H	2001/11/08	15:00:00	23	45	30	NULL	NULL	NULL
2	2	2	H	2001/11/08	16:00:00	34	67	56	NULL	NULL	NULL
3	3	3	H	2001/11/08	17:00:00	45	78	67	NULL	NULL	NULL
4	4	4	H	2001/11/08	18:00:00	56	89	70	NULL	NULL	NULL
5	5	5	H	2001/11/08	19:00:00	67	90	75	NULL	NULL	NULL

1.7.11 Helper Tables

The central data warehouse data model has the ability to separate data by customers so that each customer only has access to their own data.
This statement is based on the fact that the central data warehouse may be shared amongst customers.
It is also possible to have distinct central data warehouses for distinct customers.
IBM Tivoli Monitoring infrastructure does not provide any information about the customer_id and the host center.
IBM Tivoli Monitoring doesn't have any information regarding the hardware/software status in customer environments, and for this reason, it is completely dependent on Inventory regarding this information.
As a consequence of this, in order to provide the ability to separate the data from different databases that belong to different customers, Inventory data will be used.
The following is a possible helper table (customer_info):

Cust_db-name
Cust_db-version
customer-name
hosting-center-name

This table should contain all the databases discovered and will have unknown or default values for the customer-name and hosting-center-name.
Customers will then have the ability to fill this in by their own means.
It is recommended that a spreadsheet application (for example Excel) be used to update the customer name and hosting center name.
This database may then be re-imported into the helper table.
This information should be extracted and should be updated in the central data warehouse database accordingly.

1.7.12 Exception Tables

No Exception Tables are provided.