IBM Tivoli Monitoring, Version 5.1.1 Warehouse Enablement Pack: Implementation Guide

Version 1.1.0

SC09-7785-01

Edition notice

First Edition

Copyright Notice

Trademarks

IBM, the IBM logo, Tivoli, , Tivoli Enterprise, and TME are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

Microsoft, Windows, and Windows NT are 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.

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

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

Notices

References in this publication to Tivoli Systems or IBM products, programs, or services do not imply that they will be available in all countries in which Tivoli Systems or IBM operates. Any reference to these products, programs, or services is not intended to imply that only Tivoli Systems or IBM products, programs, or services can be used. Subject to valid intellectual property or other legally protectable right of Tivoli Systems or IBM, any functionally equivalent product, program, or service can be used instead of the referenced product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by Tivoli Systems or IBM, are the responsibility of the user. Tivoli Systems or IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, New York 10504-1785, U.S.A.

ISO 9001 Certification

This product was developed using an ISO 9001 certified quality system.

Certification has been awarded by Bureau Veritas Quality International (BVQI) (Certification No. BVQI - 92086 / A).

BVQI is a world leader in quality certification and is currently recognized by more than 20 accreditation bodies.

Contents

IBM Tivoli Monitoring, Version 5.1.1 Warehouse Enablement Pack Implementation Guide

Edition notice

1 About this document

This document describes the Warehouse Enablement Pack for IBM® Tivoli Monitoring, Version 5.1.1. It covers the following topics:

- Installing and configuring the warehouse pack
- The data flow and data structures used by the warehouse pack

With this warehouse pack, you can extract data from the IBM Tivoli Monitoring, Version 5.1.1 Middle Layer Repository (otherwise known as RIM Database) and load it into the Tivoli Enterprise Data Warehouse Common Repository. This warehouse pack contains a generic ETL1 process that is driven by the metadata provided by the Tivoli applications, that leverage on the new IBM Tivoli Monitoring Infrastructure to store historical data into a central repository (the RIM Database). This warehouse pack (otherwise known as Generic ETL1) is a prerequisite for all the applications based on IBM Tivoli Monitoring, Version 5.1.1 that are interested in collecting data for the Tivoli Enterprise Data Warehouse.

1.1 Related documentation

You can access many Tivoli® publications online using the Tivoli Information Center, which is available on the Tivoli Customer Support Web site:

http://www.tivoli.com/support/documents/

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

- Tivoli Enterprise™ Data Warehouse
- IBM DB2, DB2 Data Warehouse Center, and DB2 Warehouse Manager

The following sections list and briefly describe these libraries.

1.1.1 Tivoli Enterprise Data Warehouse

The following Tivoli Enterprise Data Warehouse documents are available on the Tivoli Enterprise Data Warehouse Documentation CD:

- *Tivoli Enterprise Data Warehouse Release Notes*, GI11-0857

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

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

  Describes how Tivoli Enterprise Data Warehouse fits into your enterprise, explains how to plan for its deployment, and gives installation and configuration instructions. It provides an introduction to the built-in program for creating and running reports, and contains maintenance procedures and troubleshooting information.

- *Enabling an Application for Tivoli Enterprise Data Warehouse*, GC32-0745

  Provides information about connecting an application to Tivoli Enterprise Data Warehouse. This book is for application programmers who use Tivoli Enterprise Data Warehouse to store and report on their application's data, data warehousing experts who import Tivoli Enterprise Data Warehouse data into business intelligence applications, and customers who use their local data in the warehouse.

1.1.2 IBM DB2, DB2 Data Warehouse Center, and DB2 Warehouse Manager

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. The DB2 library is available on the Tivoli Customer Support Web site. After you install DB2, its library is also available on your system.

The following DB2 documents are particularly relevant for people working with Tivoli Enterprise 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, 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 Data Warehouse Center.

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

  Provides the information 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 a platform-specific DB2 client. Once the DB2 client is installed, you then configure communications for both the client and server, using the DB2 GUI tools or the Command Line Processor. 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 Data Warehouse Center, and describe the actions you should take.

2 Overview

The following sections provide an overview of Tivoli Enterprise Data Warehouse and the IBM Tivoli Monitoring 5.1.1 warehouse pack.

2.1 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.

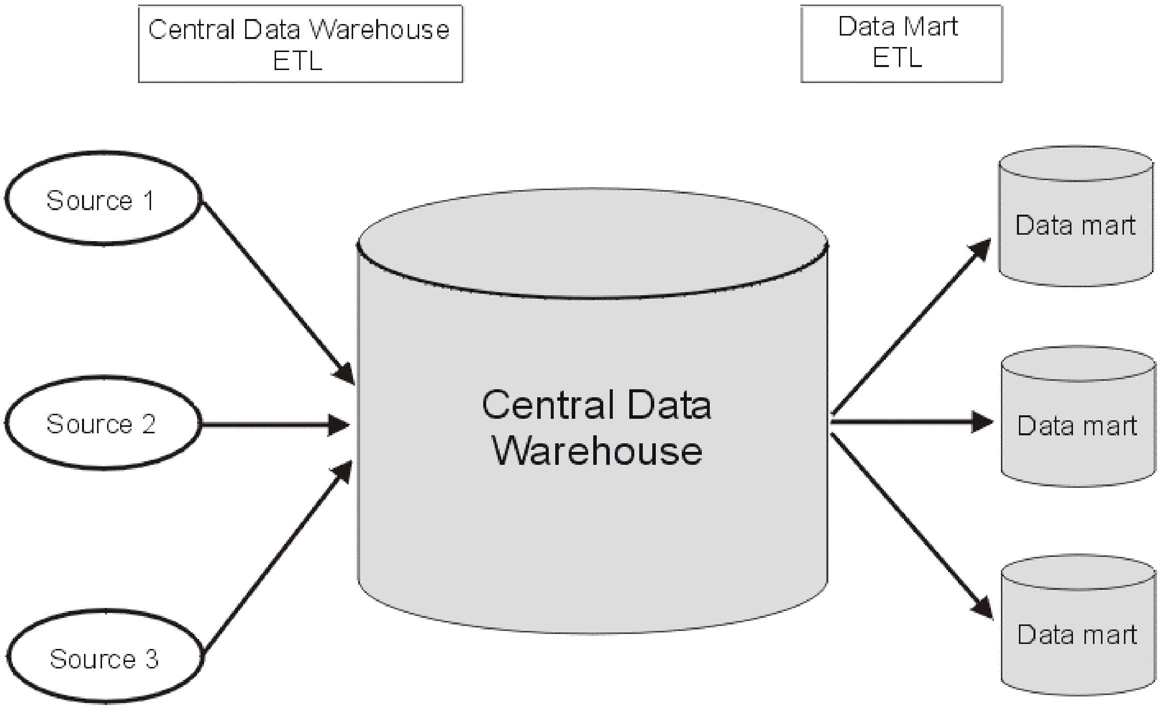

Figure 1. Tivoli Enterprise Data Warehouse overview

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.

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.

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 *warehouse pack*.

2.2 Overview of IBM Tivoli Monitoring warehouse pack

The IBM Tivoli Monitoring, Version 5.1.1 warehouse pack is shipped starting from version 5.1.1 of IBM Tivoli Monitoring and can be found under the directory tedw_apps_etl and the sub-directory AMX of the installation media.

It is intended as a general-purpose tool to perform the extraction, transformation and loading of data from the IBM Tivoli Monitoring, Version 5.1.1 Middle Layer Repository (otherwise known as RIM Database) to the Tivoli Enterprise Data Warehouse common schema. To accomplish this task, it needs the metadata (data about data) of the specific monitoring application that is logging measurements into the Middle Layer Repository; in other words, the Generic ETL1 is driven by the metadata provided by the Warehouse Enablement Packs of each monitoring application.

With this approach, it is possible to schedule always the execution of the same process, regardless of how many monitoring applications are logging data into the Middle Layer Repository, but at least one Warehouse Enablement Pack for an enabled monitoring application must be installed within the Generic ETL1.

3 Installing and configuring

3.1 Prerequisites

Before installing the IBM Tivoli Monitoring, Version 5.1.1 warehouse pack, the following software must be installed:

- IBM DB2 Universal Database Enterprise Edition Version 7.2
- IBM DB2 Universal Database Enterprise Edition Version 7.2 Fix Pak 10a
- Tivoli Enterprise Data Warehouse Version 1.1
- Tivoli Enterprise Data Warehouse 1.1 Fix Pack 3 (1.1-TDW-FP03)

You can obtain the Tivoli Enterprise Data Warehouse Fix Pack from the Tivoli Enterprise Data Warehouse Web site (http://www.ibm.com/software/sysmgmt/products/support/TivoliEnterpriseDataWarehouse.html). Click the Downloads link in the self-help section.

Refer to the Tivoli Enterprise Data Warehouse Fix pack 3 documentation for instructions on obtaining IBM DB2 Universal Database Enterprise Edition Version 7.2 Fix Pak 10a.

3.2 Supported hardware and software

The IBM Tivoli Monitoring, Version 5.1.1 warehouse pack supports all the databases supported by IBM Tivoli Monitoring, Version 5.1.1, as documented in the *IBM Tivoli Monitoring User's Guide,* Version 5.1.1, as sources of data. However, it supports IBM DB2, Version 7.2, only as a target, because this is the only database actually supported by the Tivoli Enterprise Data Warehouse (see the previous section for versioning information).

3.3 Limitations

The IBM Tivoli Monitoring, Version 5.1.1 warehouse pack leverage on the metadata provided by the application specific warehouse pack, so that, if new or different metrics are to be loaded into the Tivoli Enterprise Data Warehouse, using the Generic ETL1 process, a manual update of the application specific metadata file (xxx_cdw_data.sql, where xxx is the AVA code of the application) is required in order to describe the new metrics and how they should be organized into the warehouse data model; no automated tool is provided to perform the metadata extraction from the IBM Tivoli Monitoring Resource Models. For more detailed instructions on what tables needs to be updated, and with which data, in order to have new metrics loaded into the Tivoli Enterprise Data Warehouse, refer to section 9.6 “AMX translation and conversion tables”

For other considerations about the maximum time interval during which the collected data is available at the endpoints or gateway before being lost, if no collecting request is performed or the Middle Layer Database (otherwise known as RIM Database) is unavailable, see the IBM Tivoli Monitoring, Version 5.1.1 documentation.

3.4 Database sizing considerations

The Generic ETL1 process, provided with the IBM Tivoli Monitoring, Version 5.1.1 warehouse pack, consists of four steps and ends with inserting data at least into the TWG.Comp and TWG.Msmt, tables of the Tivoli Enterprise Data Warehouse common schema. During their execution, these steps use several temporary tables, created and destroyed as needed, so it might be appropriate to size the temporary tablespace for the TWH_CDW database.

Because the IBM Tivoli Monitoring, Version 5.1.1 warehouse pack is a prerequisite component, any other database sizing consideration is documented as part of the application that uses it to load data into the Tivoli Enterprise Data Warehouse common repository.

3.5 Data sources and targets

The following warehouse sources and targets are created by the IBM Tivoli Monitoring, Version 5.1.1 warehouse pack:

- AMX_TWH_CDW_Source: a source for the TWH_CDW database
- AMX_ITM_RIM_Source: a source for the IBM Tivoli Monitoring RIM Database
- AMX_TWH_CDW_Target: a target for the TWH_CDW database

Before running the Generic ETL1 process you must check or update the login information for all the warehouse sources and targets according to your environment. The *Installing and Configuring Tivoli Enterprise Data Warehouse: Reference Guide* contains detailed instructions about performing this task. Please note that the minimum version of the ODBC drivers that must be used for the Warehouse Source "AMX_ITM_RIM_Source" is labeled "DataWHSE 3.6.0 32-bit *RDBMS*" where *RDBMS* can be either of the following:: "INFORMIX", "Oracle8 Server", "SQL Server" and "Sybase".

3.6 Pre-installation steps

You should install the IBM Tivoli Monitoring, Version 5.1.1 warehouse pack using the same user as the one that was used to install the Tivoli Enterprise Data Warehouse core product.

If you installed the Tivoli Enterprise Data Warehouse core product using one user (for example, db2admin) and then try to install the IBM Tivoli Monitoring, Version 5.1.1 warehouse pack using another user, you will need to create a user temporary tablespace for the Warehouse Enablement Pack (WEP) install user. You must do this for the TWH_CDW database and the TWH_MART database.

To do this, use the following commands:

```
db2 "connect to TWH_CDW user <WEP install user> using <password>"
db2 "create user temporary tablespace usertmp2 managed by system using ('usertmp2')"
db2 "connect to TWH_MART user <WEP install user> using <password>"
db2 "create user temporary tablespace usertmp3 managed by system using ('usertmp3')"
```

To verify if the new tablespace has been created, try to declare a temporary table in both the databases TWH_CDW and TWH_MART, while connected as the Warehouse Enablement Pack install user:

```
db2 "declare global temporary table t1 ( c1 char(1) ) with replace on commit preserve rows not logged"
```

If you are installing the IBM Tivoli Monitoring, Version 5.1.1 warehouse pack in a Tivoli Enterprise Data Warehouse distributed environment, you must verify that the TEDW001 and TEDW002 nodes exists in the database node directory. You also need to check that the TWH_CDW database is attached to the node TEDW001 and that the TWH_MART database is attached to the node TEDW002. If the nodes do not exist or if the databases are not attached to the correct nodes, the install may fail during the execution of the after scripts.

In order to check if the TWH_CDW and TWH_MART databases are attached to the correct node, and eventually move them to the right one, you can use the following DB2 commands:

`db2 uncatlog db <database name>`	uncalog a database
`db2 uncatalog node <node-name>`	uncatalog a node
`db2 catalog db <db_name> as <alias_name> at node  <node_name>`	recatalog a database to a node
`db2 list node directory`	display the list of nodes
`db2 list db directory`	display the list of databases
`db2 catalog node tcpip TEDW001 remote <hostname hosting the TWH_CDW database> server 50000`	recreate the TEDW001 node

```
db2 catalog node tcpip TEDW002 remote <hostname hosting the
TWH_MART database> server 50000
```

recreate the TEDW002 node

Note that after each catalog/uncatalog operation you must issue a `db2 terminate` command to make the changes effective.

3.7 Installation procedure

To install the IBM Tivoli Monitoring, Version 5.1.1 warehouse pack, perform the following steps:

1. Make sure that Tivoli Enterprise Data Warehouse, Version v1.1.0 and all the prerequisite fix packs and e-fixes are installed. For instructions about installing Tivoli Enterprise Data Warehouse, see *Installing and Configuring Tivoli Enterprise Data Warehouse*.
2. Perform any pre-installation steps as described in “Pre-installation steps.”
3. Make sure you perform the installation of the warehouse pack on the right system. Typically it is the same system where the Tivoli Enterprise Data Warehouse control server is installed, even if the central data warehouse database (TWH_CDW) is located on a remote system.
4. Install the warehouse pack as described in *Installing and Configuring Tivoli Enterprise Data Warehouse*.
5. Optionally, install language support for the warehouse pack as described in *Installing and Configuring Tivoli Enterprise Data Warehouse*.
6. Perform the post-installation steps described in “Post-installation steps.”

3.8 Post-installation steps

After installing the IBM Tivoli Monitoring, Version 5.1.1 warehouse pack, using the DB2 Warehouse Control Center graphical interface, update the warehouse sources (AMX_xx_Source) and targets (AMX_xx_Target). This involves the following actions:

- Changing the user ID. You do this if db2admin is not the user that has been chosen to install the Tivoli Enterprise Data Warehouse when the three databases TWH_CDW, TWH_MD, and TWH_MART, and the RIM Database used by the IBM Tivoli Monitoring Data Collector were created.

  When you have completed this task, check that the contained tables use the right schema. That is, that they use the same schema as the new user ID you have defined for the warehouse sources and targets. For example, if you updated the user ID from db2admin to db2, then the schema for the tables contained into the warehouse source or target must also be db2. If it is not, then you must update it on the properties page for each table.

- Setting the correct passwords for the users you have entered.

3.9 Advanced configuration of the “Generic ETL1”

If you have the IBM Tivoli Monitoring, Version 5.1.1 Middle Layer Repository (the RIM Database) on a UNIX machine, where the TWH_CDW and the TWH_MART DB2 databases are installed, and you are using a Windows NT server as the warehouse center server, you can complete the installation of the IBM Tivoli Monitoring, Version 5.1.1 warehouse pack in either of the following ways:

- Use the Tivoli Enterprise Data Warehouse install shield with one unique Warehouse Agent on the Windows NT machine. Then, each time you do an extraction from the RIM Database to the TWH_CDW database, the data is transferred through the Windows NT machine, even though all the databases are on the same UNIX machine, but with an obvious degradation of the overall performances.

- Create a Warehouse Agent on the UNIX, DB2 machine (see Appendix D of *Installing and Configuring Tivoli Enterprise Data Warehouse* for a detailed explanation about creating, configuring and using a Remote Warehouse Agent). Using a Remote Warehouse Agent, you need not transfer the data through the Windows NT machine at each run of the extraction process.

When using this last configuration, the Tivoli Enterprise Data Warehouse install shield does not complete the task for you. When you have completed the installation of the "Generic ETL1" on the Windows NT warehouse center server, you must still complete the following steps manually:

- Copy the %TWH_TOP_DIR%\apps\amx tree under the $TWH_TOP_DIR/apps/amx directory on the UNIX machine.
- Copy the AMX_Extract.sh , AMX_Create_Comp_Msmt.sh and AMX_Rollup.sh script files from the $TWH_TOP_DIR/apps/amx/misc directory to $TWH_TOP_DIR/tools/bin directory on the UNIX machine, preserving the case.
- Create the temporary tablespace on the UNIX machine. You must do this because the following script did not run on the UNIX machine:

```
db2 “connect to TWH_CDW user <WEP install user> using <WEP install user password>”

db2 “create bufferpool buf8k size 500 pagesize 8 K”

restart  DB2 issuing a db2stop force command followed by a db2start command

db2 “create system temporary tablespace tssystemp8k pagesize 8 K managed by system using
('tssystemp8k') bufferpool buf8k”
```

4 Uninstalling

When you uninstall the IBM Tivoli Monitoring, Version 5.1.1 warehouse enablement pack, you do the following:

1. Remove all the tables under the schema AMX in the Common Data Warehouse (TWH_CDW) and Mart database (TWH_MART)
2. Remove all the AMX Process, AMX sources and AMX Targets in the Warehouse Control Center. This cleans particular tables in the Control Database (TWH_MD).
3. Remove all the directory structure AMX under the directory <TWH Install directory>\apps

Uninstalling the IBM Tivoli Monitoring Version 5.1.1 warehouse enablement pack has no effect on the RIM source database.

To uninstall the IBM Tivoli Monitoring, Version 5.1.1 warehouse enablement pack you must use the twh_app_deinstall.sh script, located under the %TWH_TOPDIR%\install\bin directory

However, perform this task only after you have removed from the system all the application specific warehouse packs, which depend on the Generic ETL1 process to load their data into the Tivoli Enterprise Data Warehouse common repository, with the appropriate uninstalling procedure (refer to the Tivoli Enterprise Data Warehouse documentation for details about uninstalling a warehouse pack). If you forget to uninstall some of the warehouse packs, the twh_app_deinstall.sh script will perform anyway a check for the existence of dependent application specific warehouse packs, aborting the uninstall process if some of them is still installed.

5 Maintaining

5.1 Backing up and restoring

You should plan to create a backup of the three databases used by the Tivoli Enterprise Data Warehouse (TWH_CDW, TWH_MART, and TWH_MD) on a regular basis. A sample script (dbrest.sh) in the <TWH Install directory>\apps\amx\v511\misc\tools directory shows how to restore these three databases on a Windows NT/2000 system.

The TWH_CDW database is the only one used by the "Generic ETL1" process in order to keep track of the data extracted from the RIM Database and to store additional data to be used during the reconstitution of the data hierarchy to be inserted into the Tivoli Enterprise Data Warehouse common repository, so al least this database should be backed-up; but, because the IBM Tivoli Monitoring, Version 5.1.1 warehouse pack also uses the TWH_MD database, in order to store the login information for both the warehouse source and target, performing a backup of this database, immediately after the post installation step described before, ensure to save all the configuration changes.

It is important to note that, because of the mechanism used to control the data extraction, if you restore an old backup of the RIM Database, you should update the TWG.Extract_Control table (in the TWH_CDW database) according to the maximum values of the INSERT_SEQ field that you have for all the tables defined in the RIM Database. If you don't perform this task, you will not be able to see any new data loaded into the Tivoli Enterprise Data Warehouse common repository until the INSERT_SEQ field for each table of the RIM Database, will become greater then the corresponding value defined into the TWG.Extract_Control table.

5.2 Pruning

The AMX_c10_Prune_Process prunes the IBM Tivoli Monitoring Middle Layer Repository (RIM Database) of any data older than 10 days.

The 10 days age parameter is set into the table AMX.Prune_Control. The column Age_Days is set by default to 10 but you can change it if necessary.

The prune process searches in the METRICSDATA and CATEGORIESDATA tables for data older than 10 days, and deletes in the other tables (METRICS, RMPROFILES, CATEGORIES, RESOURCES) all the rows related to this old data. That is, all the records that are no longer referenced by the METRICSDATA and/or CATEGORIESDATA tables.

The AMX.Prune_Log table contains the following four columns:

- **Prune_Action_DtTm**. This is the date and time when the pruning was performed
- **Age_Days**. This is what was inserted into the AMX.Prune_Control table (data older than Age_Days is deleted)
- **Prune_Key**. This is equal to today minus the prune Age_Days converted to number of seconds since "the epoch" (01/01/1970 00:00:00).
- **Prune_Before_DtTm**. This is the today's date minus the prune Age_Days.

Whatever the scheduling for the Generic ETL1 process, the Prune process may be added as the last step of the Generic ETL1 process, so that it is not necessary to create a separate scheduling for it.

5.3 Other tools

Some sample tools are provided under the <TWH Install directory>\apps\amx\v511\misc\tools directory. These tools are provided *as is* and are intended only as a base from which to develop more complex and system-specific tools.

- The **<Product_code>_AMX_Reset_Extract_Window.bat** script resets the Extract_Control window for IBM Tivoli Monitoring, Version 5.1.1 for the application with the product code <product-code>.

Note: This tool should be used only to restart the Extract Control window for the AMX_c05_ETL1_Process from the beginning. If you want to reset the window to the last extract, use the extract_log to get the last values of each extract function used by the Generic ETL1.

- The **List_AMX_Exceptions.bat** script runs the SQL script that lists the contents of all the AMX exception tables. These tables are filled in during the AMX_ETL1_Process when metadata do not match the data extracted from the RIM database.
- The **Extract_Win.bat** script shows the TWG.Extract_Control and TWG.Extract_Log windows based in the integer sequence (intseq). It also reformats the columns to make the output more readable.
- The **rewind_extract.sql** script updates the TWG.Extract_control window for the AMX entries to the values of a previous day. The dates in the script should be updated according to the values of the column extlog_done_dttm found in the TWG.Extract_log. This script allows to rewind the extract_control window. This can be useful when database has been restored to a previous day.

6 ETL processes

The IBM Tivoli Monitoring, Version 5.1.1 warehouse pack defines the following processes into the DB2 Warehouse Control Center graphical interface.

6.1 AMX_c05_ETL1_Process

The AMX_c05_ETL1_Process extracts data from the IBM Tivoli Monitoring Middle Layer Repository, transforms it and loads it into the Tivoli Enterprise Data Warehouse Common Repository. This process should be run once a day.

This process has the following steps:

- AMX_c05_s005_Pre_Extrat Step

  This step drops and creates staging tables that will be used during the extract step (AMX_c05_s010_RIM_Extract). You should not rerun this step before the AMX_c05_ETL1_Process has been executed successfully. If you rerun this step before the successful end of the AMX_c05_ETL1_Process, all data that has been extracted will be erased in the staging tables and nothing will be inserted in the TWH_CDW database.

- AMX_c05_s010_RIM _Extract Step

  This step creates dynamically the AMX_c05_s030_extract scripts with the suffixes of all the RIM database vendor supported (db2, oracle, informix, mssql, sybase). Each time a new warehouse enablement package is installed or removed, the script is regenerated.

  This step performs the inflow of new data from the RIM Database to some staging tables in the TWH_CDW database. Only the data from the RIM database that have their corresponding application warehouse enablement package installed will be extracted. Once the data is extracted, the TWG.Extract_Control table keeps a trace of what has been extracted by application. If the extract step fails, you should fix the problem and rerun the AMX_c05_ETL1_Process starting at the AMX_c05_s010_RIM_Extract step.

- AMX_c05_s020_Parsing Step

  This step parses the INSTANCE_KEY field in the AMX.Stage1_Instances table to get a row per Parent Key property in a new AMX.Stage_Key_Parsed table.

- AMX_c05_s030_Exception Step

  This step applies the exception rules. Go through each instance key, each attributes and each metric names to see if they are pre-defined in the metadata tables in the TWH_CDW repository and throws an exception, i.e. inserts a row into an exception table, if not.

  The exception tables are:

  - AMX.stage_attrtyp_transl_ex: an exception occurs when:
    - Some attributes of a component type are not declared in the metadata table AMX.AttrTyp_Transl.
    - A component having a unique key (not a composite key) has this key declared as an attribute and it should not.
    - The IBM Tivoli Monitoring attribute name that is part of a composite key is not the same in the metadata tables AMX.CompTyp_Transl and the AMX.AttrTyp_Transl.
  - AMX.stage_comptyp_transl_ex: an exception occurs when there is a component type in the TWG.RelnRul that do not exist in the AMX.CompTyp_Transl table.
  - AMX.stage_inst_comp_pac_ex: an exception occurs when the IBM Tivoli Monitoring instance key in the AMX.stage_key_parsed contains at least one key that is not pre-defined in the metadata for that application.

  - AMX.stage_categories_ex: an exception occurs when properties that are inside the categories table are not declared as attributes for that component type in the metadata table AMX.AttrTyp_Transl.
  - AMX.stage_metrics_ex: an exception occurs when metric names in the IBM Tivoli Monitoring METRICS table are not defined in the metadata table TWG.MsmtTyp.
  - AMX.stage_relnrul_transl_ex: an exception occurs when in the AMX.RelnRul_Transl table there is a source component type that does not correspond to a component type in the metadata table TWG.CompTyp table.
  - AMX.stage_resource_transl_ex: an exception occurs when:
    - In the AMX.stage1_resources table, there is no corresponding component type code in the table TWG.CompTyp.
    - The AMX.RelnRul_Transl table contains a relation type different from PROXY, PCHILD, SAME
    - A component source type without a host for parent has a PROXY relationship.
  - AMX.stage_rmprofiles_ex: an exception occurs when there is in the source RMPROFILES table, a category that does not correspond to a category in the metadata table AMX.Category_Transl table.

  The AMX_c05_s030_Exception Step also transforms the IBM Tivoli Monitoring formatted data into Tivoli Enterprise Data Warehouse formatted data, using the AMX Translation Tables. It also prepares some tables that will be used in the next step, such as new IBM Tivoli Monitoring staging tables with no exception data.

- AMX_c05_s040_Comp_Msmt_Step

  This step creates dynamically the amx_c05_s040_comp_msmt.db2script that consists of inserting the components, attributes, relationships, and measurements, for each warehouse enabled monitoring application, in the TWG.Comp, TWG.CompAttr, TWG.CompReln and TWG.Msmt tables. This step then executes the amx_c05_s040_comp_msmt.db2script.

  Note: If you change some metadata, you will need to rerun the Exception and the Comp_Msmt steps to get a new and correct amx_c05_s040_comp_msmt.db2 script.

7 Command line

The **AMX_c05_ETL1_Process.bat** script is a wrapper script for the AMX_c05_ETL1_Process.sh shell script, located in the <TWH install dir>/apps/amx/v511/misc/tools directory, that can be executed to run the **AMX_c05_ETL1_Process** from the command line.

This script requires that the < TWH install dir>/tools/bin directory is included in the system path, so that the bash shell and the execsql execution engine can be executed from any other system directory.

During its execution, the following error messages might be displayed:

- Error in the pre_extract step
- Error in the RIM_Extract Step
- Error in the Parsing Step
- Error in the Exception Step
- Error in the Create Comp Msmt Step

You can check the corresponding log files in the <DB2 install directory>/logging directory.

8 Audit trail

To trace the AMX_c05_ETL1_Process, see the following log files located in the DB2 log directory (for instance C:\SQLLIB\Logging)

amx_c05_s010_rim_extract.log: trace file for the Extract step

amx_c05_s020_parsing.log: trace file fro the Parsing step

amx_c05_s030_exception.log: trace file for the Exception step

amx_c05_s040_comp_msmt_diag1: trace file for the creation of the amx_c05_s040_ comp_msmt.db2 script. When no application warehouse pack has been installed within the IBM Tivoli Monitoring, Version 5.1.1 warehouse pack, this file should be empty. Each time a new application warehouse pack is installed, or each time some metadata in the application warehouse pack is changed, the amx_c05_s040_comp_msmt_diag1 file is recreated. At the same time, the amx_c05_s040_ comp_msmt.db2 script stored in <TWH install dir>/apps/amx/v511/etl/sql is recreated.

amx_c05_s040_comp_msmt_sqltemp.db2: temporary file used to store an SQL statement during the creation of the amx_c05_s040_ comp_msmt.db2 script.

amx_c05_s040_comp_msmt_sqltemp1.out: temporary file used to store the result of an SQL statement during the creation of the amx_c05_s040_ comp_msmt.db2 script.

amx_c05_s040_comp_msmt_diag2: trace file that exists only when the ETL1 Process is run at least twice. It indicates that the metadata has not changed and it keeps the time when the AMX_c05_s040_Comp_Msmt step was run.

In order to trace the AMX_c10_Prune_Process, you can look at the following log files located in the DB2 log directory (for instance C:\SQLLIB\logging):

amx_c10_s010_rim_prune.log: trace file for the prune step.

9 Generic schema implementation

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

Shaded columns in the following tables can be translated by the application. *Installing and Configuring Tivoli Enterprise Data Warehouse* contains instructions for installing support for additional languages.

Note: The IBM Tivoli Monitoring, Version 5.1.1 warehouse pack, as a prerequisite for each enabled monitoring application warehouse pack, does not directly save data into the warehouse repository, except the data used for its own functionalities.

9.1 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**
IP_HOST	NULL	IP Host	2002-07-03-20.23.04.016000	9999-01-01-12.00.00.000000
IP_INTERFACE	NULL	IP Interface	2002-07-03-20.23.04.026001	9999-01-01-12.00.00.000000
TME_ENDPOINT	NULL	Tivoli Endpoint	2002-07-03-20.23.04.036001	9999-01-01-12.00.00.000000

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

9.2 Component relationship type (table RelnTyp)

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

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

9.3 Attribute type (table AttrTyp)

AttrTyp_Cd **CHAR(17)**	**AttrTyp_Nm**[*] **VARCHAR(120)**
AMX_EID	IBM Tivoli Monitoring Endpoint Identifier
AMX_GMT_OFFSET	IBM Tivoli Monitoring Endpoint GMT Offset
CONTACT	Contact Name
DESCRIPTION	Description
LAST_IP_ADDRESS	Last IP Address
MACHINE_MODEL	Machine Model Number
MACHINE_TYPE	Machine Type
MAJOR_VERSION	Major Version Number
MANUFACTURER	Manufacturer
MINOR_VERSION	Minor Version Number

AttrTyp_Cd **CHAR(17)**	**AttrTyp_Nm*** **VARCHAR(120)**
NAME	Name
OS_NAME	Operating System Name
OS_TYPE	Operating System Type
OWNER	Owner
SERIAL_NUMBER	Serial Number
SUB_VERSION	Sub Version Number
TME_LABEL	Tivoli Endpoint Label
TME_OBJECT_ID	Tivoli Object ID
VERSION	Version Number

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

9.4 Attribute rule (table AttrRul)

CompTyp_Cd **CHAR(17)**	**AttrTyp_Cd** **CHAR(17)**	**AttrRul_Strt_DtTm** **TIMESTAMP**	**AttrRul_End_DtTm** **TIMESTAMP**	**AttrRul_Dom_Ind** **CHAR**
IP_HOST	CONTACT	2002-07-03-20.23.05.528001	9999-01-01-12.00.00.000000	N
IP_HOST	LAST_IP_ADDRESS	2002-07-03-20.23.05.418000	9999-01-01-12.00.00.000000	N
IP_HOST	MACHINE_MODEL	2002-07-03-20.23.05.568001	9999-01-01-12.00.00.000000	N
IP_HOST	MACHINE_TYPE	2002-07-03-20.23.05.558001	9999-01-01-12.00.00.000000	N
IP_HOST	MAJOR_VERSION	2002-07-03-20.23.05.478001	9999-01-01-12.00.00.000000	N
IP_HOST	MANUFACTURER	2002-07-03-20.23.05.548000	9999-01-01-12.00.00.000000	N
IP_HOST	MINOR_VERSION	2002-07-03-20.23.05.498000	9999-01-01-12.00.00.000000	N
IP_HOST	OS_NAME	2002-07-03-20.23.05.458000	9999-01-01-12.00.00.000000	N
IP_HOST	OS_TYPE	2002-07-03-20.23.05.468000	9999-01-01-12.00.00.000000	N
IP_HOST	SERIAL_NUMBER	2002-07-03-20.23.05.588000	9999-01-01-12.00.00.000000	N
IP_HOST	SUB_VERSION	2002-07-03-20.23.05.508000	9999-01-01-12.00.00.000000	N
IP_HOST	TME_LABEL	2002-07-03-20.23.05.438001	9999-01-01-12.00.00.000000	N
IP_HOST	TME_OBJECT_ID	2002-07-03-20.23.05.428000	9999-01-01-12.00.00.000000	N
IP_HOST	VERSION	2002-07-03-20.23.05.518000	9999-01-01-12.00.00.000000	N
IP_INTERFACE	CONTACT	2002-07-03-20.23.05.698001	9999-01-01-12.00.00.000000	N
IP_INTERFACE	MACHINE_MODEL	2002-07-03-20.23.05.748000	9999-01-01-12.00.00.000000	N
IP_INTERFACE	MACHINE_TYPE	2002-07-03-20.23.05.728001	9999-01-01-12.00.00.000000	N
IP_INTERFACE	MAJOR_VERSION	2002-07-03-20.23.05.658000	9999-01-01-12.00.00.000000	N
IP_INTERFACE	MANUFACTURER	2002-07-03-20.23.05.718000	9999-01-01-12.00.00.000000	N
IP_INTERFACE	MINOR_VERSION	2002-07-03-20.23.05.668000	9999-01-01-12.00.00.000000	N
IP_INTERFACE	OS_NAME	2002-07-03-20.23.05.628000	9999-01-01-12.00.00.000000	N
IP_INTERFACE	OS_TYPE	2002-07-03-20.23.05.638001	9999-01-01-12.00.00.000000	N
IP_INTERFACE	SERIAL_NUMBER	2002-07-03-20.23.05.758000	9999-01-01-12.00.00.000000	N
IP_INTERFACE	SUB_VERSION	2002-07-03-20.23.05.678001	9999-01-01-12.00.00.000000	N
IP_INTERFACE	TME_LABEL	2002-07-03-20.23.05.608001	9999-01-01-12.00.00.000000	N
IP_INTERFACE	TME_OBJECT_ID	2002-07-03-20.23.05.598001	9999-01-01-12.00.00.000000	N
IP_INTERFACE	VERSION	2002-07-03-20.23.05.688001	9999-01-01-12.00.00.000000	N
TME_ENDPOINT	CONTACT	2002-07-03-20.23.05.858001	9999-01-01-12.00.00.000000	N
TME_ENDPOINT	MACHINE_MODEL	2002-07-03-20.23.05.898001	9999-01-01-12.00.00.000000	N
TME_ENDPOINT	MACHINE_TYPE	2002-07-03-20.23.05.888000	9999-01-01-12.00.00.000000	N

CompTyp_Cd CHAR(17)	AttrTyp_Cd CHAR(17)	AttrRul_Strt_DtTm TIMESTAMP	AttrRul_End_DtTm TIMESTAMP	AttrRul_Dom_Ind CHAR
TME_ENDPOINT	MAJOR_VERSION	2002-07-03-20.23.05.808001	9999-01-01-12.00.00.000000	N
TME_ENDPOINT	MANUFACTURER	2002-07-03-20.23.05.878000	9999-01-01-12.00.00.000000	N
TME_ENDPOINT	MINOR_VERSION	2002-07-03-20.23.05.818001	9999-01-01-12.00.00.000000	N
TME_ENDPOINT	OS_NAME	2002-07-03-20.23.05.778001	9999-01-01-12.00.00.000000	N
TME_ENDPOINT	OS_TYPE	2002-07-03-20.23.05.798000	9999-01-01-12.00.00.000000	N
TME_ENDPOINT	SERIAL_NUMBER	2002-07-03-20.23.05.918000	9999-01-01-12.00.00.000000	N
TME_ENDPOINT	SUB_VERSION	2002-07-03-20.23.05.838000	9999-01-01-12.00.00.000000	N
TME_ENDPOINT	TME_LABEL	2002-07-03-20.23.05.768001	9999-01-01-12.00.00.000000	N
TME_ENDPOINT	VERSION	2002-07-03-20.23.05.848000	9999-01-01-12.00.00.000000	N

9.5 Component measurement

9.5.1 Measurement group type (table MGrpTyp)

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

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

9.5.2 Measurement group (table MGrp)

MGrp_Cd CHAR(6)	MGrpTyp_Cd CHAR(6)	MGrp_Parent_Cd CHAR(6)	MGrp_Nm* VARCHAR(120)
AVL	CATEG	NULL	Availability
MEMORY	CATEG	PERF	Memory
NETWK	CATEG	PERF	Network
PERF	CATEG	NULL	Performance
STATE	CATEG	NULL	Percentage State Measurements
STORAG	CATEG	NULL	Storage
UTIL	CATEG	NULL	Utilization
AVG_E	GROUP	NULL	Average Value Exists
MAX_E	GROUP	NULL	Maximum Value Exists
MIN_E	GROUP	NULL	Minimum Value Exists
TOT_E	GROUP	NULL	Total Value Exists

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

9.5.3 Measurement unit category (table MUnitCat)

MunitCat_Cd CHAR(6)	MunitCat_Nm VARCHAR(120)
TM	Time Duration
QTY	Quantity
PRC	Percentage
RT	Rate

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

9.5.4 Measurement unit (table MUnit)

MUnit_Cd CHAR(6)	MUnitCat_Cd CHAR(6)	Munit_Nm VARCHAR(120)
PRC	PRC	Percentage
Bps	RT	Bytes per Second
MBps	RT	Megabytes per Second
KBps	RT	Kilobytes per Second
Rps	RT	Requests per Second
Qps	RT	Quantity per Second
Qpm	RT	Quantity per Minute
QTY	QTY	Quantity
GB	QTY	Gigabytes
KB	QTY	Kilobytes
MB	QTY	Megabytes
B	QTY	Bytes
MSec	TM	Milliseconds
Sec	TM	Seconds
Min	TM	Minutes
Hr	TM	Hours
Day	TM	Days
HSc	TM	Hundredths of a Second

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

9.5.5 Time summary (table TmSum)

The period over which a measurement may be summarized.

TmSum_Cd CHAR	TmSum_Nm VARCHAR(120)
H	Hourly
D	Daily
W	Weekly
M	Monthly
Q	Quarterly
Y	Yearly

9.5.6 Measurement source (table MSrc)

MSrc_Cd CHAR(6)	MSrc_Parent_Cd CHAR(6)	MSrc_Nm VARCHAR(120)
Tivoli	NULL	Tivoli Application
AMX	Tivoli	IBM Tivoli Monitoring

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

9.6 AMX translation and conversion tables

To modify or create a resource model, and the corresponding Tivoli Enterprise Data Warehouse static data, you must create your own metadata file, which is loaded into the TWH_CDW common schema before the Generic ETL1 process is run.

This process provides a flow for the new metric to be collected into the Tivoli Enterprise Data Warehouse Common Repository by the Generic ETL1 process, according to the template provided by one of the warehouse enablement packs (for example, the IBM Tivoli Monitoring for Operating Systems warehouse pack).

You must then create the most appropriate hierarchy for the Common Data Model and map all the IBM Tivoli Monitoring specific resources and metrics to the Tivoli Enterprise Data Warehouse components and measurement types.

When you have completed this task, you can exploit the following Translation and Conversion tables.

9.6.1 RelanRul translation (table RelnRul_Transl)

This table must be filled according to the following rules:

- The CompTyp_Target_Cd always represents the Remote component in the PROXY and SAME relation.
- The CompTyp_Source_Cd and the CompTyp_Target_Cd are always components that have a host for a parent in the PROXY relation.
- All the children of a remote component should have a PCHILD relation.

CompTyp_Source_Cd* VARCHAR(17)	CompTyp_Target_Cd* VARCHAR(17)	RelnTyp_Cd* CHAR(6)	RelnRul_Strt_DtTm* TIMESTAMP	RelnRul_End_DtTm TIMESTAMP

** Primary key of the table.*

9.6.2 Resource translation (table Resource_Transl)

This table must be filled according to the following rules:

- The resource name is sometimes, but not necessarily, equivalent to the CIM class name. It should always indicate the name of the resource and should always correspond to a Tivoli Enterprise Data Warehouse component.
- In the resource translation table, you can have a one-to-one relationship or a many-to-one relationship but never a many-to-many relationship.
- Only a component of level 1 (its parent is a host) can be a remote component.

Resource **VARCHAR(128)**	**CompTyp_Cd**[*] **VARCHAR(17)**

** Primary key of the table.*

9.6.3 Category translation (table Category_Transl)

This table must be filled according to the following rules:

- The Category should contain the string defined as a Category for all the resource models of the same monitoring application.
- The MSrc_Cd should contain the product code (internally referenced as AVA code) of the monitoring application. Because of the foreign key constraint to the MSrc _Cd field of the TWG.MSrc table, the MSrc_Cd used in this table must be already defined into the TWG.MSrc table.

Category **VARCHAR(128)**	**MSrc_Cd**[*] **CHAR(6)**

** Primary key of the table.*

9.6.4 Component type (table CompTyp_Transl)

This table must be filled according to the following rules:

- MSrc_Cd should contain the product code (internally referenced as AVA code) of the monitoring application. Because of the foreign key constraint on the MSrc_Cd field of the TWG.MSrc table, the MSrc_Cd used in this table must be already defined into the TWG.MSrc table.
- ITM_Key_Property should contain the list of the key properties (long property name) that are defined in the CIM class for a component. Keys are separated by a semi-colon. All properties should be prefixed by the resource name.
- CompTyp_Cd should contain the Tivoli Enterprise Data Warehouse name of the component.
- Comp_Format_Nm should contain the list of the key properties (long property name) that make the name of the component. Keys are separated by a semi-colon.

MSrc_Cd[*] **CHAR(6)**	**ITM_Key_Property** **VARCHAR(1096)**	**CompTyp_Cd**[*] **VARCHAR(17)**	**Comp_Format_Nm** **VARCHAR(254)**

** Primary key of the table.*

9.6.5 Attribute translation (table AttrTyp_Transl)

This table must be filled according to the following rules:

- MSrc_Cd should contain the product code (internally referenced as AVA code) of the monitoring application. Because of the foreign key constraint on the MSrc _Cd field of the TWG.MSrc table, the MSrc_Cd used in this table must be already defined into the TWG.MSrc table.

- The ITM_Attr_Property should contain the properties for all the CIM classes per monitoring application.
- The AttrTyp_Cd should contain the Tivoli Enterprise Data Warehouse attribute name (17 characters maximum).

MSrc_Cd* **CHAR(6)**	**ITM_Attr_Property*** **VARCHAR(254)**	**AttrTyp_Cd*** **VARCHAR(17)**

** Primary key of the table*

9.6.6 Categories convert (table Categories_Convert)

In the IBM Tivoli Monitoring warehouse pack, the category should represent the product and should be unique, while the IBM Tivoli Monitoring Resource Models allow the definition of different categories. This table is used to map the different categories used by IBM Tivoli Monitoring into a unique category to be used in the Tivoli Enterprise Data Warehouse, to represent the source of data.

Category_Convert VARCHAR(128)	**Category VARCHAR(128)**	**Description VARCHAR(128)**

9.6.7 Instance keys convert (table Inst_Key_Convert)

The IBM Tivoli Monitoring infrastructure allows multiple entries in the instances table that should map to one unique CompTyp_Cd in Tivoli Enterprise Data Warehouse. The Generic ETL1 expects to translate a ITM_Key_Property into a Tivoli Enterprise Data Warehouse CompTyp_Cd in the AMX.CompTyp_Transl table only one way. This table is therefore used to indicate which Key_Typ should be translated in the AMX.Stage_Key_Parsed table and to which string.

Key_Typ_Convert VARCHAR(254)	**Key_Typ VARCHAR(254)**	**Description VARCHAR(128)**

9.6.8 Resources convert (table Resources_Convert)

The IBM Tivoli Monitoring infrastructure allows multiple names for the same resource that should map to one unique CompTyp_Cd in TEDW. The Generic ETL1 expects to translate an IBM Tivoli Monitoring Resource into a TEDW CompTyp_Cd in the AMX.Resource_Transl table only one way. This table is therefore used to indicate which Resource name should be translated in the AMX.Stage_Resources table and to which string.

Resource_Convert VARCHAR(128)	**Resource VARCHAR(128)**	**Description VARCHAR(128)**

9.7 AMX Helper Tables

To assign warehouse components to customers and centers, the ITM v5.1.1 Warehouse Enablement Pack provides two lookup tables AMX.CUST_LOOKUP and AMX.CENTR_LOOKUP. These two tables serve to map a hostname to a Customer or Center.

In order to use these tables, you will need first to populate the table TWG.CUST and TWG.CENTR with the name of the customer and the name of the center as well as the time zone, the geographic area, the customer identifier and the center code. Then you will need to associate a machine (hostname) to the corresponding customer identifier and center identifier in the lookup tables (AMX.CUST_LOOKUP, AMX_CENTR_LOOKUP).

Once you done this, you can then run the AMX_IBM_Tivoli_Monitoring_v5.1.1 process to collect the data from the ITM database. The endpoints found in this database will be created as component types: IP_HOST (if it was a fully qualified hostname), or <product_code>_HOST if not, and they will automatically belong to the correct Customer Identifier and Center Code.

Refer to the section Customers and Centers in the Warehouse documentation "*Enabling an Application for Tivoli Data Warehouse*".