

IBM WebSphere Transformation Extender



# Trading Manager

*Version 7.8*

**Note**

Before using this information, be sure to read the general information in "Notices" on page 65.

**30 June 2006**

This edition of this document applies to IBM WebSphere Transformation Extender Trading Manager Version 7.8; and to all subsequent releases and modifications until otherwise indicated in new editions.

To send us your comments about this document, email [DTX\\_doc\\_feedback@us.ibm.com](mailto:DTX_doc_feedback@us.ibm.com). We look forward to hearing from you.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 2006. All rights reserved.

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

---

# Contents

<b>Chapter 1. Introduction</b>	<b>1</b>
IBM WebSphere Transformation Extender Packs for EDI	2
IBM WebSphere Transformation Extender with Launcher	2
<b>Chapter 2. Installation requirements</b>	<b>3</b>
Minimum system requirements	3
System requirements for Partner Manager	3
UNIX platform	3
Prerequisites	3
Installation checklists	4
Launcher	4
<b>Chapter 3. Installing Trading Manager</b>	<b>5</b>
Determining the Trading Manager configuration	5
Installing Trading Manager on a single workstation	5
Installing Trading Manager on separate workstations	5
Installing only Partner Manager	6
Determining the ODBC driver	6
Defining an ODBC data source name	6
Resolving ODBC difficulties	6
Installation checklists	7
Installing components	7
Installing Trading Manager on Windows	7
Installing Trading Manager on UNIX	7
Removing Trading Manager components	8
Using Trading Manager with HIPAA	8
<b>Chapter 4. Configuring Trading Manager</b>	<b>9</b>
Before you begin	9
SQL scripts installed by Trading Manager	9
Creating the Trading Manager database	10
Using an Access database	10
Creating a Microsoft SQL Server Database	10
Creating an Oracle Database	11
Creating a Sybase Database	12
Creating an IBM DB2 Database	12
IBM DB2 stored procedure enhancements	13
Configuring Partner Manager	15
Import structure validation	15
Importing an X12 structure validation file	15
Importing an EDIFACT structure validation file	16
<b>Chapter 5. Installing and using Message Manager</b>	<b>17</b>
Installing Message Manager	17
Where to install Message Manager	17
General configuration and setup	18
Starting Message Manager	18
Message Manager deployment tasks	18
Step 1. Define the server types in the msg_mgr.msd file	19
Step 2. Modify the msg_mgr.msd default pathname settings	19
Step 3. Deploying Message Manager on UNIX	20
Step 4. Modify the E-mail alert settings	21
Step 5. Configure the Launcher service startup parameters	22
Step 6. Modify the Message Manager database/query file	22

Step 7. Identify the Trading Manager database. . . . .	23
Step 8. Define Partners, Post Offices and Trade Links . . . . .	23
Step 9. Run the EDI Wizard . . . . .	24
Step 10. Enable stampandsortFTP, stampandsortVAN, and stampandsortEmail maps . . . . .	24
Step 11. Modify Ecmall.mtt to support proprietary data. . . . .	24
Step 12. Add application maps and link to Message Manager. . . . .	24
Step 13. Use the deploy facility in the Integration Flow Designer . . . . .	25
Step 14. Build and copy application and communication maps . . . . .	26
Step 15. Start the Launcher service. . . . .	27
Using Message Manager . . . . .	27
Message Manager subsystems . . . . .	28
Archive subsystem . . . . .	29
EDIFACT subsystem . . . . .	29
EDIFACT inbound subsystem . . . . .	29
EDIFACT outbound subsystem . . . . .	30
StampAndSort subsystem . . . . .	30
TRADACOMS subsystem. . . . .	36
TRADACOMS inbound subsystem . . . . .	36
TRADACOMS outbound subsystem . . . . .	36
Value Added Tax (VAT) reports. . . . .	37
Update subsystem . . . . .	37
X12 subsystem . . . . .	39
X12 inbound subsystem . . . . .	39
X12 outbound subsystem . . . . .	40
<b>Chapter 6. Support for proprietary data types . . . . .</b>	<b>41</b>
Inbound data flow . . . . .	41
Adding proprietary data types . . . . .	41
To add support for a new data type . . . . .	41
To add application processing for proprietary data . . . . .	42
<b>Chapter 7. Using alias files in maps . . . . .</b>	<b>43</b>
Defining alias name and user defined fields . . . . .	43
Making the alias data available to application maps . . . . .	43
Choosing the alias file to use . . . . .	44
Alias files type tree (Alias.mtt) . . . . .	44
Creating user type trees . . . . .	44
Overview . . . . .	44
<b>Chapter 8. Checklists . . . . .</b>	<b>47</b>
Server installation checklist for microsoft SQL server . . . . .	47
Client installation checklist for Microsoft SQL Server . . . . .	48
Server installation checklist for Oracle database . . . . .	48
Client installation checklist for Oracle Server . . . . .	49
Server installation checklist for Sybase . . . . .	49
Client installation checklist for Sybase . . . . .	50
Server installation checklist for IBM DB2 database . . . . .	50
Client installation checklist for IBM DB2 Server . . . . .	51
<b>Chapter 9. Implementing custom X12 versions . . . . .</b>	<b>53</b>
Overview . . . . .	53
Type trees . . . . .	53
Partner Manager. . . . .	54
Message Manager . . . . .	54
Creating structure records . . . . .	54
TPEC to Trading Manager conversion . . . . .	55
Overview . . . . .	55
TPECAddressBook.xml file . . . . .	55
TPECPostOffices.xml file . . . . .	56
TPECTradelinks.xml file . . . . .	56

TPEC2CMConfig.txt file . . . . .	57
Running the TPEC to Trading Manager . . . . .	58
Setting up the TPEC2CM conversion map . . . . .	59
Running the "ConvertTPEC2CM" executable map . . . . .	60
Running the Partner Manager XML Autoload Utility . . . . .	60
Warnings and error messages . . . . .	61
<b>Notices . . . . .</b>	<b>65</b>
Programming interface information . . . . .	67
Trademarks and service marks . . . . .	67
<b>Index . . . . .</b>	<b>69</b>



---

## Chapter 1. Introduction

Trading Manager is a B2B management environment that uses the IBM WebSphere Transformation Extender to integrate trading partner transactions with enterprise applications. It includes two components: Partner Manager, a graphical administration component, and Message Manager, a runtime subsystem that uses trading profiles maintained by Partner Manager to automate document exchange among Electronic Data Interchange (EDI) partners. This environment consists of the following components:

- **Partner Manager** - a graphical user interface (GUI) application used to create a comprehensive database of an organization's electronic commerce information and to manage its e-commerce activity. Partner Manager is used to define your trading relationships with your e-commerce partners. It also maintains the information for each trading partner including address information, contact persons, and the individual organizational entities with which you trade business information. Partner Manager allows for multiple databases to be used or multiple copies may also access a single database.
- **Message Manager** - a pre-defined system of maps specifically designed to process electronic commerce data. The executable maps are used for validation and routing of data.
- **Design Studio** - used to create maps that perform enterprise application integration by converting data from one form to another (typically to integrate e-commerce data with applications) and routing that information from its source to its target. The Design Studio features include the Integration Flow Designer, Map and Type Designers, Metadata Type Importers for XML, EDI, and DBMS applications, and prebuilt adapters for messaging and transporting.
- **Integration Flow Designer** - used to define and organize maps into logical groups or comprehensive workflow systems. The Integration Flow Designer is used by a system analyst or system administrator responsible for linking internal applications with the Message Manager components provided as part of Trading Manager. Using the information defined in Partner Manager, the Integration Flow Designer is used to tailor the Message Manager for your organization's needs and to define run time settings for execution on the Launcher. It is a component of the Design Studio.
- **Launcher** - used to execute the Message Manager maps. It is installed from the IBM WebSphere Transformation Extender. The Launcher can run on a single server or across multiple servers to distribute processing. Run-time event triggers are configured graphically using the Integration Flow Designer. During execution, data flows seamlessly between system components.
- **Resource Adapters** - provide connectivity to a wide array of types of data sources and targets such as databases, messaging systems, FTP, or e-mail.

Electronic commerce data flows from external sources to internal destinations and from internal sources to external destinations, and possibly to other internal organizations. Trading Manager manages this flow and the tracking of e-commerce data.

---

## IBM WebSphere Transformation Extender Packs for EDI

The IBM WebSphere Transformation Extender Packs for EDI are collections of type trees representing the different versions of an EDI standard. Integration packages for the following EDI standards are available:

- X12
- EDIFACT
- TRADACOMS
- ODETTE
- EANCOM

---

## IBM WebSphere Transformation Extender with Launcher

The IBM WebSphere Transformation Extender with Launcher provides the server with components to execute the Launcher. It includes the Launcher, the Management Tools (for Windows 2000 and Windows XP only), and the Command Server.

The IBM WebSphere Transformation Extender with Launcher must be installed before deploying the Message Manager system. The IBM WebSphere Transformation Extender with Launcher helps you automate the execution of system maps. This map execution is based on time or source events, in contrast to the Command Server that executes maps according to commands that you supply involving only one map at a time.



---

## Chapter 2. Installation requirements

These instructions describe the requirements for installing Trading Manager. The database associated with Trading Manager can reside on the same machine as the IBM WebSphere Transformation Extender with Launcher or on another network-accessible machine.

The Design Studio must be installed prior to installing Trading Manager.

---

### Minimum system requirements

The following documentation describes the requirements for Partner Manager, as well as the Windows 2000, Windows XP, UNIX, and Linux platforms.

#### System requirements for Partner Manager

The minimum requirements to install and run Partner Manager are:

- Microsoft Windows 2000, or Windows XP.
- 128 MB RAM, however, 256 MB RAM is recommended

#### UNIX platform

The Launcher for UNIX (or Linux) is a multi-threaded daemon that runs on UNIX platforms.

The **cmsvu** and **cmsvc** shared libraries for each UNIX operating system are installed in the *install\_dir\tmgr\_vn.n\platform\_support* directory. Before starting Message Manager on the server, ensure that the correct **cmsvu** and **cmsvc** file is copied to the **libs** directory located in the IBM install directory on the UNIX server.

Refer to the Launcher documentation for UNIX installation and configuration information.

---

### Prerequisites

- **E-mail** - Ensure that connectivity and access to e-mail service and a user account for Message Manager is in place. Message Manager supports MS Mail, MS Exchange, Lotus CCMail, Lotus Notes and SMTP/POP3. Message Manager uses the e-mail service to send dynamic problem alerts to individuals responsible for managing the electronic commerce program.
- **Trading partner requirements** - To analyze your requirements, consider the following:
  - Number of trading partners to be supported
  - Versions of standards and EDI documents (810, 850, 997, and others) to be exchanged
  - Application format specifications for any internal applications interfacing with the EDI system
  - Communication requirements for transporting data between application systems and the Trading Manager server. Requirements include the type of communications to use (e-mail, FTP, messaging product, and others), userids, passwords, naming conventions, and responsible parties

---

## Installation checklists

Before installing Trading Manager, please complete the installation checklist for your database software. See the Checklists documentation for more information.

---

## Launcher

The Launcher is installed as part of the IBM WebSphere Transformation Extender. It should already be installed. If it is not installed, refer to the documentation for installation instructions.

---

## Chapter 3. Installing Trading Manager

These instructions describe the installation of the Trading Manager software components.

If you have an earlier version of Trading Manager installed, delete the **MessageManager.msl** file located in the *install\_dir*\systems directory. If you do not delete this .msl file, the system will fail. For further information; the new default filename is **msg\_mgr.msl**.

Refer to the Launcher documentation for installation information.

After the Launcher and Partner Manager are installed, you can configure the system as described in the documentation titled "Configuring Trading Manager".

HIPAA customers should refer to the documentation titled: "Using Trading Manager with HIPAA" .

---

### Determining the Trading Manager configuration

Trading Manager enables you to manage your installation depending on your e-commerce processing needs. You can install all Trading Manager components on a single machine, or assign Trading Manager functions to different and dedicated machines.

#### Installing Trading Manager on a single workstation

On a single workstation running all Trading Manager client/server components including the database, the following components must be installed:

- Server
- Partner Manager
- Message Manager
- Database Scripts

**Note:** Ensure that you have installed the Design Studio and IBM WebSphere Transformation Extender first.

#### Installing Trading Manager on separate workstations

If you are installing Trading Manager on separate workstations and servers in a client/server configuration, install these components:

- **Client workstation:** Partner Manager and Message Manager
- **Server machine where the Launcher is also installed:** Server; this installation provides the Launcher and directory structure required for Message Manager map execution.
- **Database Server:** Database scripts

## Installing only Partner Manager

On individual workstations using only Partner Manager to create information about trading and application partners, post offices, etc., only the Partner Manager application need be installed.

---

### Determining the ODBC driver

Before you use Partner Manager, you must configure an open database connectivity (ODBC) data source name (DSN) on each client machine that will run Partner Manager. The DSN associates an ODBC driver with your database type. Review the Database Requirements to determine the ODBC driver to select before defining your DSN. Ask your Database Administrator to assist you. Then proceed with defining the DSN.

---

### Defining an ODBC data source name

After you determine the ODBC driver to use, complete the following steps to define the DSN.

To define an ODBC DSN perform the following steps:

1. From the **Start** menu select:  
**Settings** → **Control Panel** → **Administrative Tools** → **Data Sources (ODBD)**  
The **ODBC Data Source Administrator** displays.
  2. Select the **System DSN** tab (not **User DSN**).  
Selecting **System DSN** enables you to create a DSN that is visible to all users; selecting **User DSN** creates a DSN that is only visible to the specified user on a particular machine. Do not create a **User DSN**.
  3. Click **Add**.  
The Create a New Data Source Name window opens.
  4. Select the appropriate driver for your database from the list.
  5. Click **Finish**.  
The ODBC Setup window opens.
  6. Enter the driver-specific setup parameters.  
The name you assign in the Data Source Name window entry may be any name. However, IBM recommends that you use **Trading\_Manager\_TEST** (for testing) and **Trading\_Manager\_PROD** (for production) to simplify support and consulting consistency. The name entered will also be the selection for your DSN when you configure Partner Manager. You may enter an optional description in this window.
  7. Complete the window as required by the driver provider.
  8. Click **OK**.
  9. The name you assigned appears in the **Name** list in the **ODBC Data Source Administrator**.
  10. Click **OK**.
- 

### Resolving ODBC difficulties

If you experience difficulties with the ODBC connection when first installing or running Partner Manager, get the latest ODBC drivers from your database vendor. Ensure the driver is for your version of the database and client operating system.

Oracle users: ODBC driver version 8.1.7.8.0 or better is required, depending on your Oracle Configuration. See the Oracle Metalink article titled "Supported ODBC Configurations"; document ID 66403.1.

Contact your Database Administrator for a password to view this article.

[http://metalink.oracle.com/metalink/plsql/ml2\\_documents.showDocument?p\\_database\\_id=NOT&p\\_id=66403.1](http://metalink.oracle.com/metalink/plsql/ml2_documents.showDocument?p_database_id=NOT&p_id=66403.1)

In the event this link is out of date, search MetaLink for "Supported ODBC Configurations".

Database vendors supply free ODBC drivers and they are available for download from their web sites. For example:

- Microsoft: [www.microsoft.com/data](http://www.microsoft.com/data). Select **Downloads** and get the latest MDAC installation available.
- Oracle: <http://otn.oracle.com/software/content.html>

For others, search your vendor's web site for **ODBC** or **ODBC Download**.

---

## Installation checklists

Before installing the Trading Manager, please complete the installation checklist for your database software. See "Checklists".

---

## Installing components

Before you begin installation, close any other programs running on your computer. Having other programs running may cause installation errors.

Before installing Trading Manager, the Design Studio must be already installed. If you have not done so already, perform this installation first by inserting the Design Studio CD into the CD-ROM drive and follow the installation instructions for installing the Design Studio.

You can optionally install the IBM WebSphere Transformation Extender for system execution on the local Windows 2000, or Windows XP client workstation using the Launcher. It is available on a separate IBM WebSphere Transformation Extender CD that installs the Launcher and related tools.

## Installing Trading Manager on Windows

To install the Trading Manager components on Windows 2000 and XP perform the following steps:

1. Insert the Trading Manager CD into your CD-ROM drive.
2. If the program does not automatically start, go to the **Start** menu and select **Run**.
3. Enter **d:\setup.exe** and click **OK** (where **d:** indicates the CD-ROM drive).
4. Follow the setup instructions to complete installation.

## Installing Trading Manager on UNIX

To install Trading Manager on HP-UX, Sun Solaris, or IBM RS/6000 AIX, or Linux:

1. Copy the **cminstall.ksh** file to the IBM WebSphere Transformation Extender install directory on the target machine.
2. Execute **cminstall.ksh** on the target machine. Refer to the Launcher documentation for more information.
3. Navigate to the platform folder and copy the shared libraries to the **libs** directory under the IBM WebSphere Transformation Extender install directory on your UNIX machine.

---

## Removing Trading Manager components

To remove Trading Manager Components perform the following steps:

1. From the **Start** menu, select **Settings > Control Panel > Add/Remove Programs**.
2. Select the product component in the list provided.
3. Click **Add/Remove**.

---

## Using Trading Manager with HIPAA

The **hipaa\_x12\_4010\_qualifier.dat** and **hipaa\_x12\_4010\_structure.dat** files must be imported into Trading Manager before any HIPAA trade links can be created. For a complete description of this activity, refer to the IBM WebSphere Transformation Extender Pack for HIPAA EDI documentation.

---

## Chapter 4. Configuring Trading Manager

These instructions describe how to configure Trading Manager with the databases, the Launcher Launcher and the Partner Manager.

How you configure Trading Manager depends on:

- The type of database your system is using: Microsoft Access, Microsoft SQL Server, Oracle, Sybase, or DB2
- Where the database is located
- Which applications interface with your system
- Whether you have a standalone or multi-machine configuration

---

### Before you begin

Before configuring Trading Manager, ensure that:

- The database administrator has completed the database installation
- A System Data Source Name (DSN) has been defined for the ODBC driver
- Trading Manager components have been installed

You may then configure the Trading Manager database, Partner Manager settings, and Message Manager. The general configuration steps are:

- Create the Trading Manager database.
- Run the SQL scripts for Microsoft SQL Server, Oracle, and Sybase to create the Trading Manager database objects.
- Configure Partner Manager to point to the Trading Manager database.
- Configure Message Manager to point to the Trading Manager database. These instructions describe how to change the database type in the *install\_dir\tmgr\_vn.n\mmgr\update.mdq* file to match the database type for your installation.

### SQL scripts installed by Trading Manager

Trading Manager installs the following SQL script files in the Trading Manager directory *install\_dir\tmgr\_vn.n\pmgr\sql*. The following descriptions apply to the scripts for Oracle, Microsoft SQL Server, Sybase, and IBM DB2.

- Installation script (*\*instnn.sql*): creates the tables, indexes, stored procedures, and views used by Partner Manager.
- Database load script (*\*loadnn.sql*): loads the database with data such as Folders, Version tables, Transaction Sets, and Functional IDs.
- Drop script (*\*dropnn.sql*): deletes all Partner Manager tables and data in the database.

Database	Script
Oracle	<b>orainstnn.sql</b> <b>oraloadnn.sql</b> <b>oradropnn.sql</b>
Microsoft SQL Server	sqlinstnn.sql sqlloadnn.sql sqldropnn.sql
Sybase	sybaseinstnn.sql sybasedropnn.sql sybaseloadnn.sql sybaseloadnn_2.sql
IBM DB2	db2instnn.sql db2loadnn.sql db2dropnn.sql

---

## Creating the Trading Manager database

See the following topics to create a database.

- "Using an Access Database"
- "Creating a Microsoft SQL Server Database"
- "Creating an Oracle Database"
- "Creating a Sybase Database"
- "Creating an IBM DB2 Database"

### Using an Access database

An Access database (**tmgrnn.mdb**) is shipped with the Trading Manager client. The Access database is installed in the *install\_dir\tmgr\_v7.8\pmgr* directory. This database can be placed locally or on a server. If located on a server, it will be necessary to map a drive to that location before configuring the ODBC DSN Connection. See the documentation titled "Installing Trading Manager" for further information.

Due to performance reasons, it is recommended that you use the Access database for test or training purposes only; it should not be used in a production environment.

### Creating a Microsoft SQL Server Database

When using Microsoft SQL Server, create the database data and log devices, and then create the Trading Manager database using these devices. The name of the database you just created cannot contain spaces or special characters. Minimum disk space for the database data device is 100 MB; a recommended size is 500 MB. Minimum disk space for the database log device is 20 MB; a recommended size is 100 MB.



If any part of this procedure needs to be repeated, use **sqldropnm.sql** to delete all Partner Manager tables and data in the database.

Use caution: the **sqldropnm.sql** routine deletes all Partner Manager tables and data, and cannot be reversed. It should be used only if the preceding steps need to be repeated.

To run the Microsoft SQL Server scripts

1. Start your **SQL Query** tool and connect to the Trading Manager database you just created.
2. Select **File** → **Open** and open the file **sqlinstnm.sql** from the default Trading Manager installation directory. If you changed the default directory during installation, use the correct directory for your install.
3. Execute **sqlinstnm.sql** and wait for the last output message to be displayed: **Install Complete**. Check the output for errors and resolve any errors before proceeding.
4. Select **File** → **Open** and open the file **sqlloadnm.sql**.
5. Execute **sqlloadnm.sql** and wait for it to complete.
6. Check the output for any errors using the scrollbar; the only output should be multiple "1 Rows Affected" messages.

## Creating an Oracle Database

When using Oracle Server, use the default ORCL instance or create an instance specifically for Trading Manager. Minimum disk space required for the instance is 100 MB. If you expect heavy transaction volume, use the auto-expand feature of Oracle.

If any part of this step needs to be repeated, use **oradropnm.sql** to delete all Partner Manager tables and data in the database. Use caution: **oradropnm.sql** cannot be reversed and should be used only if these steps must be repeated.

To run the Oracle SQL scripts

1. Log on to the desired Oracle instance in SQL\*PLUS.
2. Create a spool output by entering the following (you can change this file name, or, if available on your version of SQL\*PLUS, use the **File Spool** option to set it):  

```
spool c:\temp\inst.log
```
3. Process **orainstnm.sql** (the path for the SQL files are in the directory where you installed Partner Manager, for example):  

```
install_dir\tmgr_vn.n\pmgr\sql\orainstnm.sql
```
4. When complete, shut off spooling with **spool off** or use the **File Spool** option, if available.
5. Check the spool file created in Step 2 for errors. Resolve any errors before proceeding.
6. If there are no errors, enter **commit** in SQL\*PLUS to save the changes.
7. Create another spool output by entering:  

```
spool c:\temp\load.log
```
8. Process **oraloadnm.sql**:  

```
install_dir\tmgr_vn.n\pmgr\sql\oraloadnm.sql
```

9. When complete, shut off spooling with **spool off** or use the **File Spool** option, if available.
10. Check the spool file created in Step 7 for errors. Resolve any errors before proceeding.
11. If there are no errors, enter **commit** into SQL\*PLUS to save the changes.
12. Exit SQL\*PLUS.

## Creating a Sybase Database

When using Sybase, create the database data and log devices and then create the Trading Manager database using these devices. Minimum disk space for the database data device is 100 MB. Minimum disk space for the database log device is 20 MB.

**Note:** If any part of this step needs to be repeated, a file named **sybasedrop $nn$ .sql** can drop all the tables created by **sybaseinst $nn$ .sql**. Use caution: the **sybasedrop $nn$ .sql** routine deletes all Partner Manager tables and data and cannot be reversed. It should only be used if the preceding steps need to be repeated.

To create the Sybase Database

1. Open the SQL editing tool.
2. Select **File** → **Open** and open the installation script from the default Trading Manager installation directory. If you changed the default directory during installation, use the correct directory for your install. Select the **sybaseinst $nn$ .sql** script.
3. Execute the installation script and wait for the last output message to be displayed: Install Complete.

**Note:** **Note** On some versions of Sybase, you will receive the following warning: "Row size could exceed row size limit, which is xxxx bytes (where xxxx can vary)". You can safely ignore this warning.

4. Check the output for errors; resolve any errors before proceeding.
5. Select **File** → **Open** and open the file **sybaseload $nn$ .sql**.
6. Execute **sybaseload $nn$ .sql** and wait for it to complete.
7. Check the output for any errors.
8. Select **File** → **Open** and open the file **sybaseload $nn$ \_2.sql**.
9. Execute **sybaseload $nn$ \_2.sql** and wait for it to complete.
10. Check the output for any errors.

## Creating an IBM DB2 Database

Construct the database using scripts loaded during the Trading Manager installation. Minimum disk space required is 200 MB.

To create the IBM DB2 Database

1. Contact your DBA to get a database name, a user name, and a password.
2. Verify with the DBA that the autocommit is on. If the autocommit is not on, either issue commits manually or modify the **.sql** scripts to include the commits.

3. Open the **db2instnn.sql** file in an editor. Change the first executable line to match your database name and user/password. For example, change **sample** and **db2admin** in the following executable line:  
connect to sample user db2admin using db2admin
4. Change the tablespace name (search and replace in **DBNAME USERSPACE1**) as directed by your DBA. Trading Manager requires a DB2 tablespace page size of at least 8K; 32K is recommended. Ask your DBA to create such a tablespace, and replace USERSPACE1 with the name of this tablespace
5. Save the file.
6. Using the DB2 Command Window, or the Command Line Processor, execute the new **db2instnn.sql** saved in step 5. Recommended usage is:  
db2 -tvvf 'filepath\db2instnn.sql'>'filepath\logfile' In this example, Windows is used for the output redirection. Change that syntax to match your operating system or consider using the **-r{filename}** flag. In addition, if you wish to control auto-commit, you may add the **-c** flag to control commits (see your DB2 documentation for command line parameter information)  
The Mainframe database tool **'SPUFI'** may also be used to execute scripts.

**Note:** Your DBA will instruct you on how to run SQL scripts. The scripts supplied are semi-colon delimited plain text SQL statements.

7. After execution, review the results for errors.
8. Open **db2loadnn.sql** in an editor and change the connect statement for the database, user name, and password and then save the file.
9. Using the DB2 Command Window, or the Command Line Processor, execute the new **db2loadnn.sql** saved in step 8.
10. After execution, review the results for errors.  
If error free, the installation of **.sql** scripts is complete.

## IBM DB2 stored procedure enhancements

The following procedures install eight Java Stored Procedures and one User Defined Function (UDF) into your database. These procedures should be performed by your DB2 DBA.

DB2 Stored Procedures are not supported on DB2 USS.

Be aware that if multiple databases are in use, for example, one for test and one for production, you must repeat the steps for each database.

### Permissions

These stored procedures require access to **sysibm.sysdummy1**. Confirm the Message Manager user name (the DB2 user name running the map systems) has access to this table by issuing this select statement:

```
select * from sysibm.sysdummy1
```

If this statement reports an error, issue the following SQL statement from the DBA account:

```
grant all on sysibm.sysdummy1 to public;
```

## Software requirements

The following procedures were tested using the DB2 Development Center, version 8, fix pack 8. If you are not on fix pack 8 of your client tools, and you encounter errors during this procedure, it is suggested you obtain and install that fix pack from the IBM web site.

To install the Stored Procedures, perform the following steps:

1. Unzip the DB2 Stored Procedures file **Tmgrn.n\_DB2\_SP.zip** installed by Trading Manager. This file is located in the *install\_dir /tmgr\_n.n/pmgr/sql* directory.
2. Start the DB2 Development Center. **Start All Programs** → **IBM DB2** → **Development Tools** → **Development Center**
3. Create a new project, and add a new Database Connection for your target database. Make certain that you:
  - use the same User ID and Password for the database connection that will be used by the Message Manager user
  - perform the Test Connection option to verify the connection information is correct
  - leave the Options page blank (SQL Schema, etc.).
4. Import the 8 stored procedures. Perform the following steps for each:  
**EDF\_INCREMENTMSG.java**
5. In the **Project View** right click on **Stored Procedures**, then select **Import**.
6. Select **File System** and **Source** file for the Source to activate the **Import Wizard**.
7. Select a java stored procedure file from your work area as the Source File name (**EDF\_INCREMENTMSG.java**). Click on **Next**.
8. Click on **Next for Entry Points**.
9. Click on **Next for Parameters**. You must change each parameter to **varchar**. To do this:
  - a. Select each parameter line and then click **Change**.
  - b. Change the SQL Type field to **varchar**. Leave the length to the default (4000). Repeat for each parameter in the list.
  - c. Click **Next** once you confirm the list displayed shows all datatypes are **varchar**.
10. Enter the specific file name as the Stored Procedure Name (**EDF\_INCREMENTMSG.java**) if it is not already displayed. This is the same name as the source file, without the **.java** extension. Click on **Next**.
11. Keep the jar file name displayed. Verify the **Build** and **Replace duplicate objects in project** options are checked. Click on **Next**.
12. Verify the **Summary**. Click on **Finish**.
13. Verify the results show **Build successful** in the **Output View** of the Development Center.
14. Repeat these steps for the remaining stored procedures.

## Import the user defined function

The following procedures describe how to import the user defined function:  
**TMGR\_TS\_FMT.sql**

To import the user defined function To import the user defined function, perform the following steps:

1. In the **Project View**, right click on **User-Defined Functions**, and select **Import**.
2. Select **File System** and **Source** file for the Source to activate the Import Wizard.
3. Select the User-Defined Function sql file from your work area as the Source File name (**TMGR\_TS\_FMT.sql**). Click **Next**.
4. Click **Next** for **Entry Points**.
5. Click on Next for Parameters, leaving the defaults displayed.
6. Verify the Function Name is **TMGR\_TS\_FMT**. Click **Next**.
7. Verify that the **Build** and **Replace duplicate objects in project** options are checked. Click **Next**.
8. Verify the Summary, then click **Finish**.
9. Verify the results show **Build successful** in the Output View of the Development Center.

---

## Configuring Partner Manager

After the scripts have completed, configure Partner Manager.

To configure Partner Manager to point to the Trading Manager database

The first time that Partner Manager starts, it prompts for the DSN you configured in the documentation titled "Installing Trading Manager".

1. Start Partner Manager by selecting **Start** → **Programs** → **IBM WebSphere Transformation Extender** → **Trading Manager** → **Partner Manager**.
2. The first time Partner Manager starts, the Partner Manager Database Selection window opens.
3. Select a vendor from the **Database Vendor** drop-down list
4. Select the **Data Source Name** you created in "Installing Trading Manager" from the **DSN** drop-down list.
5. Enter **User ID** and **Password** (optional).
6. Enter an **Optional Description** if desired. This description appears on the title bar when Partner Manager is running.
7. Click **OK**.

---

## Import structure validation

For all databases created other than Access, the X12 structure validation file and the EDIFACT structure validation file must be imported.

The file need only be imported only once per database created.

To import an X12 Structure Validation file

### Importing an X12 structure validation file

To import an X12 Structure Validation File

1. Start Partner Manager with the DSN created in "Configuring Partner Manager"
2. Click **Utilities** in the tool bar. The **Utilities** navigator displays.
3. Click **Standards**.

4. In the **Standards Maintenance** window, select **X12 Structure Validation: Import** from the **X12 Standards** drop-down box.
5. Click the Find File button and select the **StructureValidation.dctl** file, which installed in the *install\_dir\tmgr\_vnn\pmgr* directory.
6. Click the **Begin Import** button to import this data.

## Importing an EDIFACT structure validation file

### To import an X12 Structure Validation File

1. Start Partner Manager with the DSN created in "Configuring Partner Manager"
2. Click **Utilities** in the tool bar. The **Utilities** navigator displays.
3. Click **Standards**.
4. In the **Standards Maintenance** window, select **EDIFACT Structure Validation: Import** from the **EDIFACT Standards** drop-down box.
5. Click the Find File button and select the **EDIFACT\_StructureValidation.dctl** file, which installed in the *install\_dir\tmgr\_vnn\pmgr* directory.
6. Click the **Begin Import** button to import this data.

---

## Chapter 5. Installing and using Message Manager

These instructions describe how to install and use the Message Manager . The Message Manager is the business logic that provides the routing, archival, standards validation, audit, and alert functions within the Trading Manager. The Message Manager runs on the Launcher and is viewed, implemented, and configured using the Integration Flow Designer. The Message Manager provides a generic electronic commerce framework with EDI-specific functions.

---

### Installing Message Manager

The Message Manager source files are installed as part of the Trading Manager. "Installing Trading Manager" describes how to install the Message Manager, which is part of Trading Manager.

#### Where to install Message Manager

The Message Manager can be installed onto a client workstation or the Trading Manager Launcher machine. The source Message Manager source maps and trees are installed within a valid Message Manager directory structure, such as *install\_dir\tmgr\_vn.n\mmgr*. There are advantages and disadvantages to each approach, which are discussed below.

##### Client workstation installation

Installing the Message Manager on a client workstation is the default option. This option requires connectivity to the IBM WebSphere Transformation Extender Launcher machine and the complete Design Studio environment. There are advantages and disadvantages to this type of installation.

As an advantage, this installation provides a single point of control for the Trading Manager operations. Also, better performance is achieved while running Design Studio as there is no network overhead.

As a disadvantage, it is difficult to share Message Manager responsibilities with other resources. Also, an extra step is required when adding new versions or transactions to the system.

##### IBM WebSphere Transformation Extender Launcher installation

Installing the Message Manager on the Trading Manager IBM WebSphere Transformation Extender Launcher machine (Windows 2000 or Windows XP platform) is also an alternative. It still requires connectivity between the client workstation of the individual administering the system and the server. This option also assumes the installation of the complete Design Studio environment on that administration client. It does not require the installation of the Design Studio components on the server. There are advantages and disadvantages to this installation.

An advantage of this installation is that it simplifies the sharing of Message Manager responsibilities due to centralized location of the source. It also eliminates the need for an extra copy step when building maps.

As a disadvantage, it may slow performance while running Design Studio clients due to network overhead. Another disadvantage is that standard drive mappings are required if multiple users are supported. For example, each client must map the server machine as Y.

---

## General configuration and setup

You must complete several activities before you can begin using Trading Manager.

- Install the Launcher Trading Manager components. See the documentation titled "Installing Trading Manager" for installation instructions.
- Complete the configuration. See "Configuring Trading Manager".
- Define trading partner information including trading relationships to the Partner Manager database using the Partner Manager client. See the Partner Manager documentation for more information.

---

## Starting Message Manager

You start the Message Manager by opening the Message Manager system definition file **msg\_mgr.msdl** in the Integration Flow Designer, installed as part of the Design Studio.

To start the Message Manager:

- From the **Start** menu, select **Programs > IBM version > Trading Manager > Message Manager**.

The Integration Flow Designer opens the **msg\_mgr.msdl** system definition file.

---

## Message Manager deployment tasks

After you have completed the Trading Manager general configuration and setup steps, you are ready to prepare the Message Manager for deployment. Deploying Message Manager consists of the following activities:

1. Define the server types in the **msg\_mgr.msdl** file.
2. Modify the **msg\_mgr.msdl** default pathname settings to match your physical environment.
3. Deploy Message Manager on UNIX.
4. Modify the email alert settings to match your email environment.
5. Configure the Launcher service startup parameters.
6. Modify the Message Manager database query file for your database type.
7. Identify the Partner Manager database in Partner Manager
8. Define Partners, Post Offices and Trade Links in Partner Manager.
9. Run the EDI Wizard.
10. Enable the **stampandsortFTP**, **stampandsortVAN**, and **stampandsortEmail** maps.
11. Modify **Eemail.mtt** to support proprietary data.
12. Add application maps and Link to Message Manager.
13. Use the Deploy Facility in the Integration Flow Designer.
14. Build and copy application and communication maps.
15. Start the Launcher.



## Step 1. Define the server types in the msg\_mgr.msd file

In the following steps, use the Integration Flow Designer to define the server types in the **MessageManager** and **RunMaps** systems.

To define the server types

1. From the **Start** menu, select **Programs IBM WebSphere Transformation Extender → Trading Manager → Message Manager**.
2. In the IFD navigator, click the **Composition** tab to show the **msg\_mgr** system definition file.
3. Select **msg\_mgr**.
4. From the **Server** menu, choose **New**.  
The Add Server window opens.
5. Enter all values and click **OK**.

## Step 2. Modify the msg\_mgr.msd default pathname settings

The Message Manager map and other components are configured assuming a default directory structure of:

```
c:\install_dir\tmgr_vn.n\mmgr.
```

If you installed Message Manager to a different location, you need to update the Message Manager path names to reflect your environment.

To update the pathnames, follow the example steps to open the **msg\_mgr.msd** system definition file in the Integration Flow Designer (IFD) and use the **Replace** feature. The **Replace** feature enables you to change the Design-Time Paths and Run-Time Paths for **msg\_mgr.msd**:

- **Design-Time Paths** are the directory paths for your system definition file and referenced source/compiled map components
- **Run-Time Paths** are the directory paths for input and output cards, audit log, trace and work files, and server map

The following example steps assume that you installed and will operate in the **d:\apps** directory and your client directory structure mirrors your server directory structure. After completing this step, your map components point to the correct path and file on your **d:\install\_dir \tmgr\_vn.n\mmgr** directory structure. You can verify this by trying to access any of the map components in Message Manager.

### To change the design-time and run-time paths

1. From the **Start** menu, select **Programs → IBM WebSphere Transformation Extender → Trading Manager → Message Manager**.
2. In the IFD navigator, click the **Composition** tab to show the **msg\_mgr** system definition file.
3. Double-click **MessageManager** to show the **MessageManager** system.
4. From the **Edit** menu, choose **More Find/Replace**.  
The Find/Replace window opens. When the **Replace** tab is selected the window opens.
5. Specify the **Find What** search criteria. To locate all path names that begin with c;, enter c: in the **Find What** field and select **Path Names** for the **Search Criteria**.

6. Click the **Settings** button to specify additional search parameters.  
The Path Name Settings window opens.
7. Ensure that **Design-Time Paths** is enabled.
8. Click **OK** to save the settings and close the Path Name Settings window.  
The Path Name Settings window closes and you are returned to the Find/Replace window.
9. With the **Replace** tab view selected, click **Find**.  
A list of the design-time paths that match the search criteria is displayed in the list. In this case, the `c:\install_dir` directory structure is found.
10. In the **Replace** field, enter the path name you wish to replace: `c:.`
11. In the **With** field, specify the actual location of your **Design-Time Paths**; in this example, `d:\apps`.
12. Click **Select All** to select all entries in the list of paths matching the search criteria.
13. Click **Replace**.
14. Repeat steps 6 through 13; in step 7, disable **Design-Time Paths** and enable **Run-Time Paths**.
15. After performing this procedure for the **MessageManager** system, repeat all steps for the **RunMaps** systems.  
Your map components now point to the correct path and file on your `d:\apps\IBM\tmgr_vn.n\mmgr` directory structure. You can verify this by trying to access any of the map components in Message Manager.

### To change the .mrn path names

1. Open the **Resource Registry**.
2. Right-click on the **Resource Files** folder and choose **Open**.
3. Navigate to the `install_dir\tmgr_vn.n\mmgr` and open the `mmgr.mrn` file.  
The file opens in the Resource Registry window.
4. Expand the `mmgr` file and expand the **Resources** folder.
5. Double-click `mmgr_share`.  
The **Edit** Resource window opens.
6. In the **Resource Value** column, enter the new location of the share directory for the required server.

For further information on the Resource Registry, see the Resource Registry documentation.

## Step 3. Deploying Message Manager on UNIX

If the target server is a UNIX based, Message Manager maps must be built and deployed to the UNIX server. To do this, the server location of all Message Manager maps, must be changed to reflect the deployment directory.

Message Manager maps use standard Windows directory nomenclature which must be changed to UNIX nomenclature (as an example, `\` replaced by `/`).

1. From the **Start** menu, select **Programs > IBM WebSphere Transformation Extender > Trading Manager > Message Manager**.
2. In the IFD navigator, click the **Composition** tab to show the `msg_mgr` system definition file.
3. Double-click **MessageManager** to show the **MessageManager** system.

4. From the **Edit** menu, choose **More Find/Replace**.  
The Find window opens.
5. Click the **Replace** tab.
6. The Replace window opens.
7. Specify the **Find What** search criteria. To locate all path names that begin with c:, enter c: in the **Find What** field and select **Path Names** for the **Search Criteria**.
8. Click the **Settings** button to specify additional search parameters.  
The Path Name Settings window opens.
9. Disable the **Design Time Paths** checkbox.
10. Enable the **Run Time Paths** checkbox.
11. Disable all checkboxes in the **Run Time Paths** area of the window except the **Audit Log** and **Server Map Location** checkboxes.
12. Click **OK**.  
The Path Name Settings window closes and you are returned to the Find/Replace window.
13. In the **Find What** field, enter `C:\install_dir`.
14. Click **Find**.
15. Select all show entries.
16. In the **With** field enter the UNIX installation directory (excluding the `mmgr_vn.n/mmgr`).
17. Repeat steps 11 to 13, changing the FIND string to `\`, and the Replace string with `/`.

## Step 4. Modify the E-mail alert settings

The Message Manager is configured with a default e-mail type of MAPI. If your mail system is other than a MAPI mail system (such as Microsoft Mail or Microsoft Exchange), modify a map rule to prepare the Message Manager for your environment.

To change the default email alert setting perform the following steps:

1. Open the **Resource Registry**.
2. Open the `mmgr.msd` and `mmgr.mrc` files.
3. Select the relevant server. For example, **LocalServer**.
4. Select the `mmgr_e-mail_settings` Resource.
5. Select the **Resource Value** and enter the override settings.
6. Turn **Encryption On** or **Off**.

**Note:** If you use passwords, you will want to use the encryption feature.

7. Click **OK**.
8. Exit out by selecting **Exit** from the **File** menu.
9. Save changes.

See the E-mail Adapter documentation for information regarding the e-mail adapter.

## Step 5. Configure the Launcher service startup parameters

The following procedure is optional and is applicable to Windows 2000 and Windows XP only.

Some e-mail options supported by Trading Manager require a modification to the default Launcher service Startup parameters. MAPI mail systems such as Microsoft Mail and Microsoft Exchange require access to a user profile when receiving MAPI calls from a service. Configuration requires the creation of a user profile under an existing user account.

It is recommended that you create an **M4EC** user with appropriate permissions including the **User Rights Policy - Log on as a service**. This permission allows the Launcher Service to be accessed by the **M4EC** userid even if the server is running in unattended Automatic mode.

After the **M4EC** userid is established, create a **Mail** profile named **M4EC** that has permission to send mail to any appropriate parties. The **M4EC** mail profile does not need to receive mail to fulfill the alert functions. If you use the pager alert option, ensure that your mail administrator has given the profile sufficient permissions.

After you or your system administrator creates the **M4EC** userid and **M4EC** mail profile, modify the default **Startup** parameters for the Launcher Service.

To modify the default Launcher service startup parameters on Windows 2000 and Windows XP perform the following steps:

1. From the **Start** menu, open the **Control Panel**.
2. Double-click **Administrative Tools**.
3. Double-click **Services**.  
The Services window opens.
4. Double-click Launcher.  
The service name may also include a version number.
5. On the **General** tab, select **Automatic** from the **Startup type** drop-down box.
6. Click the **LogOn** tab, enable the **This account** field and enter **M4EC**.
7. Enter the M4EC password in the **Password** and **Confirm password** fields.
8. Click **Apply** and then click **OK**.

## Step 6. Modify the Message Manager database/query file

You need to change the database type in the *install\_dir\tmgr\_vn.n\mmgr\update.mdq* file to match the database type for your installation, for example Microsoft SQL Server or Oracle. Trading Manager is configured to use Microsoft Access as its database. If you have built your prototype environment using Microsoft Access (as many users do), you must convert from Access to your database type in preparation for your production environment.

The Message Manager default database definition is an open database connectivity (ODBC) database for the Windows platform as shown in the following figure. The System Data Source Name (DSN) **TMgrDB** in the **Database** and **Runtime** drop-down lists in the example is for example purposes only; you create your System DSN when you use the Partner Manager interface. See the Partner Manager documentation for more information.

Each Partner Manager client must have an ODBC System data source name (DSN) for both test and production servers.

To modify the update.mdq database/query file perform the following steps:

See the Database Interface Designer documentation for information on defining databases.

1. Open the Database Interface Designer.
2. Open the **Update.mdq** file in *install\_dir\tmgr\_vn.n\mmgr*.
3. Expand **Update** to show the **TMgrDB** database definition; select **TMgrDB**.
4. From the **Database** menu, select **Edit**.  
The **Database Definition** window opens.
5. Expand the **Adapter** setting.
6. From the **Type** drop-down list, select the appropriate database type.  
The database-specific fields appear in the **Database Definition** window.
7. Specify the **Platform** from the drop-down list.
8. Enter the appropriate information for the selected database.
9. Click **OK** to set these database settings.
10. From the **File** menu, choose **Save** to save the database/query file.

## Step 7. Identify the Trading Manager database

In this step, identify the ODBC Data Source Name for each Partner Manager client. Every client must have an ODBC System data source name (DSN) for both test and production servers.

### Identify the Trading Manager database connection

It is not necessary to define an ODBC Data Source Name (DSN) for your database because you are no longer using ODBC connectivity from the Message Manager server. However, Partner Manager requires an ODBC DSN definition on every client machine from which it will be used.

To identify the ODBC DSN for each client

1. On the Trading Manager client machine, start Partner Manager.
2. From the **File** menu, choose **Database Settings** and enter the appropriate information into the fields in the Database Selection window.

## Step 8. Define Partners, Post Offices and Trade Links

Using the Partner Manager, you must define at least one of each of the following to prepare for the remaining steps:

- Internal trading partner
- External trading partner
- **GET** post office
- **PUT** post office
- Trade link

See the Partner Manager documentation for information on defining trading relationships in Partner Manager.

## Step 9. Run the EDI Wizard

After defining one or more trade links in Partner Manager, run the EDI Type Tree Wizard to create the **x12mail.mtt** and **EDIFmail.mtt** type tree required by Message Manager. See the Partner Manager documentation for instructions.

Ensure that you create or copy the **X12mail.mtt** and **EDIFmail.mtt** in the *install\_dir\tmgr\_vn.n\mmgr* directory. Message Manager uses this file for validation purposes.

## Step 10. Enable stampandsortFTP, stampandsortVAN, and stampandsortEmail maps

The following procedure is optional.

Refer to "stampandsortFTP/VAN/Email Maps" for instructions to set up FTP, VAN, and e-mail communications in the **StampAndSort** subsystem.

## Step 11. Modify Ecmail.mtt to support proprietary data

The following procedure is optional.

If you will be supporting non-EDI data types, modify the **Ecmail.mtt** type tree before building the Message Manager maps.

## Step 12. Add application maps and link to Message Manager

Message Manager does not address your organization's specific application mapping requirements. It retrieves or sends data, validates, audits, and routes it to a directory for further action; it does not transform that data without a user-written map.

The following example:

- Adds your application map that transforms an inbound X12 version 3050 Purchase Order (850) into a flat file
- Configures the Launcher settings so the map executes when an input trigger appears. You will define the input trigger to copy the flat file to a directory scanned by the Launcher.

The example assumes that this is your first user-written map in the newly-installed Message Manager directory structure.

See the Integration Flow Designer documentation for information about creating **.msd** files and adding systems, subsystems, and map components.

To add an application map

It is recommended that all application and communication maps are specified in a system definition file other than **msg\_mgr.msd**. Having a separate file allows you to manage your version control and allows you to upgrade your Message Manager at any time without affecting your specific implementation.

1. Create a new system definition file named **AppMap**.
  - Select **System Definition Files** in the navigator.

- From the File menu, choose New.
  - Right-click the new system file (**SystemFile1**) and save it as **AppMap**.
2. Add the user application map to the system and name it **POtoFlat**.
- Enable the Toolbox by choosing **Toolbox** from the **View** menu.
  - Click the **Add Source Map** tool on the Toolbox.
  - Click the **AppMap** system window.
  - In the **Add Source Map** window, browse for the **.mms** map source file to add, name it **POtoFlat**, and click **OK**.

To configure Launcher settings

You configure the Launcher settings so that the **POtoFlat** map component executes (or triggers) when the EDI data arrives in your Put File post office. Here, configure the **POtoFlat** map component to have an Input Event trigger for the first input card with the File source defined as **d:\apps\IBM\tmgr\_vn.n\mmgr\3050850\\*i.APP**

For this example, assume the following:

- The Put File PO Nickname is APP and its location is defined in Partner Manager as **d:\apps\IBM\tmgr\_vn.n\mmgr\3050850\**
- The EDI data is arriving from a VAN post office named **VAN**, which retrieves its data every hour
- As Message Manager gets the data from VAN, it passes through the **stampandsortfile**, **sortx12**, and **roux12** maps before being written to the Put File Post Office as **d:\apps\IBM\tmgr\_vn.n\mmgr\3050850\VAN00001199809150001oi.APP**
- In the **AppMap.msd** system window, ensure that the **Execution Mode** drop-down menu is set to Launcher.
- Click the execution settings button to display the Launcher Settings window.
- Click **OK** to save the Launcher settings.

## Step 13. Use the deploy facility in the Integration Flow Designer

The Integration Flow Designer documentation describes the Deploy facility in more detail.

The Message Manager is delivered as a set of source maps. To complete the deployment tasks, you may build compiled map objects, transfer them to their appropriate server directory locations, and generate the Launcher control files (**.msl**) as separate procedures using the Integration Flow Designer (IFD).

However, the IFD Deploy facility automates these build, transfer, and generate steps that prepare systems to run on designated servers. The IFD includes four Deploy scripts to automate these three separate tasks. The Deploy scripts are **DeployMessageManager**, **BuildRunMaps**, **Deploy-MRN-Update** and **GenerateMSLonly**.

The following procedures enable you to:

- Build the MessageManager and RunMaps maps
- Generate the **msg\_mgr.msl** Launcher control file

- Copy the compiled map files (**.mmc**) and Launchercontrol file (**.msl**) to the corresponding server directory (optional)

To use the Deploy facility

1. In the IFD navigator, click the **Composition** tab to show the **msg\_mgr** system definition file.
2. Expand **msg\_mgr** to show the **MessageManager** and **RunMaps** systems.
3. Right-click **MessageManager** and select **Deploy** → **Definitions**.
4. Select **Generate and Transfer Launcher Control File** and click **Details**.
5. In the window, select the path and filename to save the Launcher control file (**.msl**).  
The default full path is `c:\install_dir\systems\msg_mgr.msl`
6. Click **Save** and **Close**.
7. Right-click **MessageManager** and select **DeployMessageManager**.  
The IFD builds the **MessageManager** maps and generates the **msg\_mgr.msl** Launcher control file in the specified folder. It also copies compiled map objects (**.mmc**) to the server directory.
8. Click **OK** to view the specified deployment results.
9. Right-click **RunMaps** in the navigator and select **BuildRunMaps**.  
The IFD builds the **RunMaps** maps.
10. Click **OK** to view the deployment results.

## Step 14. Build and copy application and communication maps

The following steps can be performed in a deploy script you create. The Integration Flow Designer documentation describes the Deploy facility in more detail.

Build your application and communication maps as described in the following procedure; ensure that you select the application and communication map system definition file. All maps should compile without error. If any errors occur, those errors must be resolved before copying them to the server. Then copy the compiled map files to the corresponding server directory.

To build the application and communication system source map objects and copy the compiled map objects to the server

1. Open the system definition file (**.msd**) in the Integration Flow Designer.
2. Display the system window for the system.
3. From the **System** menu, choose **Build Maps**.
4. Click **Results** to view a detailed summary of the build process and verify that all maps have been built successfully.  
All maps should compile without error. If any errors occur, those errors must be resolved before continuing
5. Click **OK**.
6. Copy the compiled map objects (**.mmc**) to the corresponding server directory.

### Generate the Launcher control file for an application system

If you need to generate a system control file for these maps, perform the following.

To generate the Launcher control file for an application system



1. Open the application's system definition file in the Integration Flow Designer.
2. Display the system window for the system containing the application map components.  
Always generate the Launcher control file `.msl` at the top level system. For instance, if **AppSysA** contains subsystem components **AppSys1** and **AppSys2**, generate the file for **AppSysA**.
3. From the **System** menu, choose **Generate**.  
The Save As window opens to prompt you to enter the file name and location for the `.msl` file.
4. Enter the name and location for the `.msl` file:
  - Navigate to and save the `.msl` file to the **Systems** directory on the Trading Manager server.
5. Click **Save** to generate the file.

## Step 15. Start the Launcher service

The final step in deploying the Message Manager is to start the Launcher Service. The Launcher documentation contains detailed information about configuring and starting the Launcher on both the Windows and UNIX platforms.

To start the Launcher service on Windows 2000 and Windows XP perform the following steps:

1. From the **Start** menu, open the **Control Panel**.
2. Double-click **Administrative Tools**.
3. Double-click **Services**.  
The Services window opens.
4. Select **Launcher**, right-click, and select **Start**. Or, click (**Start Service**) in the tool bar.  
The service name may also include a version number.
5. When the Launcher is started, the **Status** column will display **Started**.

See the Launcher documentation for Launcher starting information.

### Launcher Manager tools

The following are the Manager Tools for the Launcher:

- Launcher Administration
- Management Console
- Launcher Monitor
- Snapshot Viewer

You can use the Management Tools for status information and debugging. The Launcher management tools are described in the Launcher documentation.

---

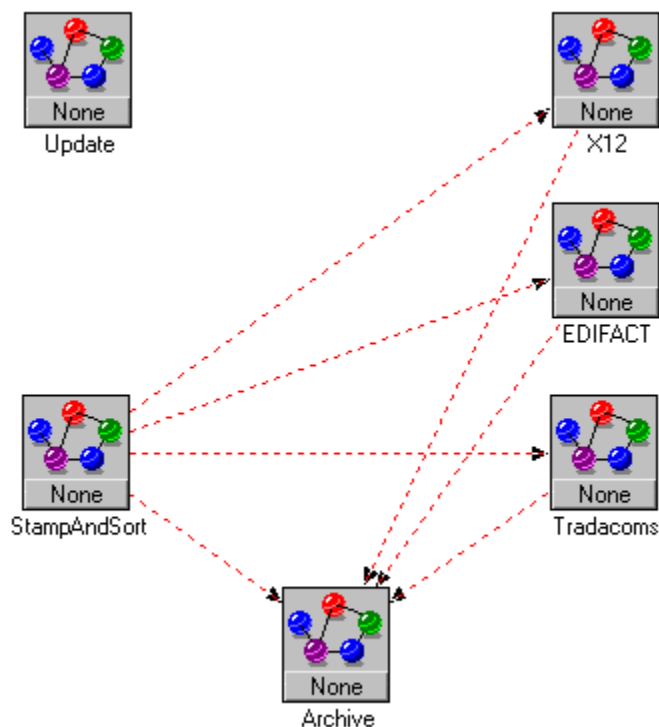
## Using Message Manager

The following topics are described here:

- The **MessageManager** system, which is defined in the `msg_mgr.msdl` system definition file
- Inbound and outbound data flows
- Subsystems in `msg_mgr.msdl`

As installed, the **MessageManager** system is a set of source maps. The system retrieves or sends data, then validates, audits, and routes it for further action. After adding your application map(s) according to your e-commerce configuration, you compile the maps and transfer them to their Trading Manager server locations before starting the Launcher. After you start the Launcher, event triggers execute the compiled maps and process your data.

The **MessageManager** system implements the logical functionality and EDI-specific features of the Trading Manager. It also provides a general framework for any data you choose to support in the system. The **MessageManager** system includes six major subsystems: **Archive**, **EDIFACT**, **StampAndSort**, **TRADACOMS**, **Update**, and **X12**. Each major subsystem contains maps or other subsystems. The following figure shows a top-level view of the **MessageManager** system.



The **StampAndSort** and **Update** subsystems provide a general data-handling framework. The **X12** subsystem is specific to ANSI X12 EDI functionality.

To fully deploy the Message Manager, a user must add:

- Application maps
- Implementation-specific configuration settings
- Trading partner information supplied through the Partner Manager
- Actual data activity (that is, events on the Launcher)

---

## Message Manager subsystems

The `msg_mgr.msd` system definition file has two executable systems: **MessageManager** and **RunMaps**. The **MessageManager** system has six subsystems: **Archive**, **EDIFACT**, **StampAndSort**, **TRADACOMS**, **Update**, and **X12**.

## Archive subsystem

The primary function of the Archive subsystem is to archive both inbound and outbound data. The Archive components will archive files processed by the system and place them into the **archive\<process>\<process>\_date.zip** file. Folder **stampfil** is used to archive all input data to Trading Manager, inbound or outbound. Folders **sendx12**, **sendedf**, and **sendtrc** are used to archive outbound files going out from Trading Manager.

You can selectively turn on/off the archiving process by going into the "Message Manager Configuration" options in Partner Manager. Options available are:

- Turn on/off archiving of Inbound files into Trading Manager.
- Turn on/off archiving of Outbound files into Trading Manager.
- Turn on/off archiving of Outbound files out from Trading Manager.

The **batchutil** component will perform batch archiving of online traffic information into the online archive. It is scheduled to run every night at 11:30 PM. Batch archiving is turned on/off and the corresponding options are specified through the "Message Manager Configuration" options in Partner Manager.

## EDIFACT subsystem

The EDIFACT subsystem contains two subsystems: **EDIFACT Inbound** and **EDIFACT Outbound**.

- The **EDIFACT Inbound** subsystem provides sorting, validating, and routing functions for data flowing from external trading partners to internal trading partners.
- The **EDIFACT Outbound** system provides enveloping, validating, and routing functions for data flowing from internal trading partners to external trading partners.

## EDIFACT inbound subsystem

The EDIFACT Inbound subsystem performs the validation and routing functions on inbound EDIFACT formatted data. The EDI data flows through three primary maps; **sortedf**, **edfctrlnb** and **edifact\_in\_alert**.

### sortedf map

This map checks the UNB and UNG (of functional group) control structures for valid trading partner information.

The UNB fields examined include:

- Interchange Sender ID qualifier
- Interchange Sender ID
- Interchange Sender Routing Address
- Interchange Received ID qualifier
- Interchange Receiver Id
- Interchange Receiver routing Address
- Test Indicator

The UNG (or functional group) fields examined include:

- Message Type
- Message Version
- Message Release

- Controlling Agency
- Associated Assigned Code

These values are compared to tradelinks defined in Partner Manager based on the GET Post Office. If no match is found and an Unknown Tradelink has not been defined, the data is rejected through the alert map.

### **edfctrlnb map**

The **edfctrlnb** map updates outbound traffic information with the status reflected in Inbound CONTRL acknowledgement.

### **edifact\_in\_alert map**

The **edifact\_in\_alert** map is a common map component used throughout the Message Manager. The **edifact\_in\_alert** map is configured to send alert notifications through e-mail or pager whenever data is rejected.

In the EDIFACT Inbound subsystem, the **edifact\_in\_alert** map is responsible for sending alert notifications when an error condition occurs in the **sortedf**, **edifact\_in\_alert** or **edfctrlnb** maps. The rejected data is written to the *install\_dir\tmgr\_vn.n\mmgr\reject* directory and the appropriate form of notification is sent to the user-specified destination.

## **EDIFACT outbound subsystem**

The EDIFACT Outbound subsystem performs the enveloping, validating and routing functions on Outbound EDIFACT formatted data. The EDI data flows through two primary maps; the **send\_problem\_data\_alerts** map and the **sendedf** map.

### **Send problem data alerts map**

The **Send Problem Data Alerts** map is configured to send alert notifications through e-mail or pager whenever data is rejected.

In the EDIFACT Outbound subsystem, the **Send Problem Data Alerts** map is responsible for sending alert notifications when an error condition occurs in the **envedf** or **sendedf** maps. The rejected data is written to the *install\_dir\tmgr\_vn.n\mmgr\reject* directory and the appropriate form of notification sent to the user-specified destination.

## **StampAndSort subsystem**

The purpose of the **StampAndSort** subsystem is to route different data types to their appropriate subsystem or map.

For example, if you retrieve a file from a customer's FTP server that contains ANSI X12 EDI data and proprietary flat file format, the **StampAndSort** subsystem regroups similar data types into a single file, then routes them to either the X12 subsystem or a user-defined map or subsystem designed to handle proprietary data.

The **StampAndSort** subsystem also includes the standard error-handling and alert mechanisms for unrecognizable data and an audit function for tracking purposes.

The **StampAndSort** subsystem serves as the entry and exit point for data flowing through Trading Manager.

The three primary functions of the **StampAndSort** subsystem are as follows:

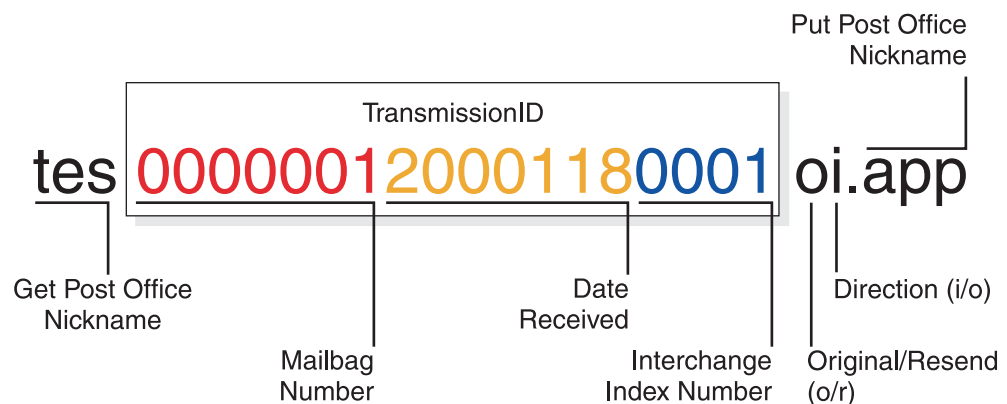
- Send and receive data
- Prepare the data for processing by assigning it a unique **ThreadID** as its filename
- Identify databases (EDIFACT, TRADACOMS, and X12)

The **StampAndSort** subsystem is shown in the following illustration:

### ThreadID definition

The Trading Manager uses the **ThreadID** to track unique data objects as they pass through the system and to facilitate concurrent processing. The elements of the **ThreadID** are described in the following figure:

The **ThreadID** contains the **TransmissionID**.



The requirements for the **ThreadID** naming conventions:

- **Get Post Office Nickname** - The first three characters of the file name are the nickname of the **Get** post office from which the e-commerce data originated.
- **Mailbag Number** - Number of different unique files received from a given post office per day.
- **Date Received** - Date the data goes through the **StampAndSort** subsystem (when **ThreadID** gets created).
- **Interchange Index Number** - A unique counter for each interchange created. This counter provides a unique **ThreadID** in support of multi-threaded execution of a user supplied application map.
- **Origin** - This value identifies whether this data is original data (**O**) going through the Trading Manager for the first time or is data that is being resent (**R**) through the Trading Manager. See the *Partner Manager Reference Guide* for information on resending data.
- **Direction** - The direction identifies whether the data is in an Inbound (**I**) or Outbound (**O**) direction.
- **Put Post Office Nickname** - The last three characters following the period are the nickname of the **Put** post office where the e-commerce data is being placed.

### Inbound data archiving

As data is passed thru the Message Manager it is broken up into the respective Interchanges and archived in a specified directory.

To specify an archived directory

1. Click the **Server Directory Configuration** button on the Utilities window.

The Server Directory Configuration Maintenance window appears.

2. Type the desired archive directory in the **Archive (contains 'in' and 'out' directories) field**.
3. Click **Save**.

If you do not specify an archive directory, you will not be able to automatically archive your files. If you do create an archive, data can be edited and resent from the Interchange Traffic.

The archive directory you specify must include **in**, **out**, and **day** directories. For example, *install\_dir\tmgr\_vn.n\mmgr\archive*.

The archived filename will be `<thread_id>_<interchange_id>.<edi_standard>`.

`<thread_id>` is the internal thread\_id of the transmission, `<interchange_id>` is the Interchange control number, and `<edi_standard>` will be EDIFACT or X12. For example, *0000010200106210000\_635.X12*.

This data is used by the resend and editing functionality in Partner Manager and can be zipped or deleted from there. See the Partner Manager documentation for instructions on how to resend data.

### **stampandsortFTP/VAN/Email maps**

The **stampandsortFTP**, **stampandsortVAN**, and **stampandsortEmail** maps are identical with the exception of the communication protocol. The configuration requirements for these three maps are described here, using the **stampandsortFTP** map as a specific example.

The **stampandsortFTP** map is designed to retrieve data using the FTP protocol and to submit that data to the Message Manager. The **stampandsortFTP** map performs the following four tasks:

- Gets data through the FTP resource adapter, using the FTP configuration information you supplied when you defined the **FTP Get** post office. An example FTP command line might be:  
-URL ftp://ediuser:password@192.1.1.50/receive.txt;type=i
- Creates a unique **ThreadID** for the entire FTP transmission (the contents are concatenated as a single data stream if you use the FTP command **MGET**).
- Archives the data retrieved through FTP in the *install\_dir\tmgr\_vn.n\mmgr\archive\stampapp* directory with an archive name adhering to the archive naming convention requirements.
- Writes the data out to a file as *install\_dir\tmgr\_vn.n\mmgr\stampec\\*.mec*. The \* represents the valid **ThreadID** assigned to the FTP transmission.

As delivered with the Message Manager, the **stampandsortFTP** map is *not* configured to run by default. The **stampandsortFTP** map does not have time or input event triggers defined. The following procedure describes how to activate the **stampandsortFTP** map.

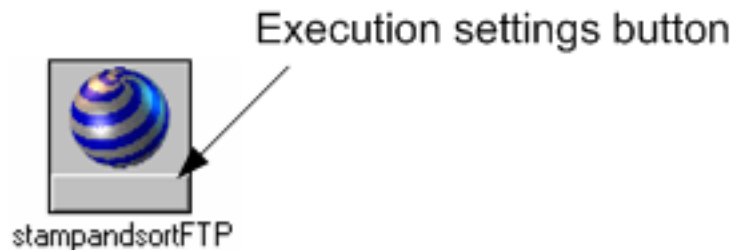
To activate the stampandsortFTP map

The following procedures assume you have experience with Partner Manager and the Design Studio.

1. Create an **FTP Get** post office in Partner Manager with a valid FTP command line.  
(Create post offices using the Partner Manager Post Office navigator. In this navigator window, click **Add**.)

See the FTP Adapter documentation for information about valid FTP command lines.

1. Open the **msg\_mgr.msd** system in the Integration Flow Designer.
2. Select the **stampandsortFTP** map component in the **StampAndSort** subsystem.
3. Click the execution settings button on the **stampandsortFTP** map component to display the Launcher Settings window.

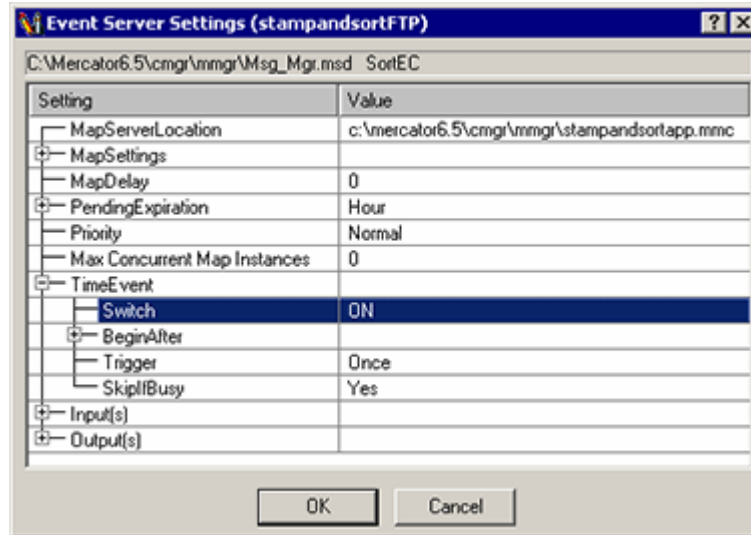


4. For the **TimeEvent** setting, add a time event trigger by setting the **TimeEvent Switch** to **ON** and specifying the **Trigger Interval** settings.

MapDelay	0
⊕ PendingExpiration	Hour
Priority	Normal
Max Concurrent Map Instances	0
⊖ TimeEvent	
Switch	ON
⊕ BeginAfter	
Trigger	Once
SkipIfBusy	Yes
⊕ Input(s)	
⊕ Output(s)	

As discussed in the **export** map documentation, it is recommended that time events for FTP sessions be scheduled to start a few minutes after system startup to ensure the export process has completed before receiving data. You may also consider distributing FTP sessions over time if your systems are bandwidth-constrained.

5. Modify the **Input(s)** information for the #1 **POInfo** input card.



The GET source field for input card 1 (POInfo) of the stampandsort maps need to be updated with a string to indicate the Post Office nickname and the direction as follows **-PON EDI -DIR i**, where PON and DIR are keywords and are case-insensitive. The value for dir can be i (inbound as shown in the preceding figure), or o (outbound).

**Note:** The three-character **PO Nickname** must match the nickname created in Partner Manager. The Message Manager uses this three-letter nickname to create a valid and unique **ThreadID**.

## Creating multiple stampandsortFTP maps

You will need an FTP session for each FTP **Get** post office that you define. You will probably need more than one FTP session to support your e-commerce needs. For each additional FTP location you want to support, create a copy of the **stampandsortFTP** map and repeat the steps in the previous procedure.

You do not need to copy the **stampandsortFTP** map into the **msg\_mgr.msd** system definition file. It is recommended that you create a separate system definition file (**.msd**) for all communication maps. Having a separate file allows you to upgrade your core Message Manager at any time without affecting your specific implementation.

To create a copy of stampandsortFTP in the Message Manager

1. Open the **msg\_mgr.msd** system in the Integration Flow Designer.
2. Select the **stampandsortFTP** map component in the **StampAndSort** subsystem.
3. From the **Edit** menu, choose **Copy**.
4. From the **Edit** menu, choose **Paste**.

The **Copy** command creates a default map component named **stampandsortFTP(2)**. The **stampandsortFTP(2)** copy is pasted on top of the source **stampandsortFTP** map. Move the **stampandsortFTP(2)** map to view both the source and the duplicate map.

5. Rename the new **stampandsortFTP(2)** map component to uniquely identify this FTP session:
  - Right-click **stampandsortFTP(2)** and select **Edit Map Component**.
  - In the **Component Name** field, enter a new name for the map component.



- Click OK.

As a reminder, the **stampandsortFTP**, **stampandsortVAN** and **stampandsortEmail** maps are identical with the exception of the communication protocol they use. To configure the **stampandsortVAN** or **stampandsortEmail** maps, follow the same steps as outlined, replacing the FTP-specific information with VAN or e-mail equivalents.

### **stampandsortfile map**

The **stampandsortfile** map is similar to the other **Stamp\*** maps in that it is responsible for creating a **ThreadID**, archiving the data, and writing the data out for retrieval by the **stampandsortfile** map. The **stampandsortfile** map is unique, however, because it requires no configuration. **stampandsortfile** is pre-configured to trigger on any file event that occurs in the installed **mail** directory specified by the **mmgr\_mail** resource name.

The only requirement for correctly implementing File Post Offices is the use of valid file names. Any e-commerce data object must have a valid **ThreadID** to successfully process inbound and outbound data objects. The creation of a valid **ThreadID** is done automatically for the application maps. For the **stampandsortfile** map, however, it is the user's responsibility to create a valid file-triggering name when writing the data to the **mail** directory. The precise file naming conventions for File Post Office files must be followed.

### **Get file post office and put file post office**

The **stampandsortfile** map is only relevant for **Get** File Post Offices. **Put** File Post Offices are implemented directly within the Message Manager and use the Partner Manager post office definitions to route data.

It is usually a good idea to create multiple physical **Put** File Post Office directories, because they perform a "mailboxing" function as well as a triggering function. When Message Manager receives inbound data, that data is validated and routed to the **Put** post office specified in the trade link. This **Put** post office is usually defined as a File Post Office, although it can be any of the post office types (even VAN). User written application maps are then configured to trigger on the arrival of data in a particular **Put** File Post Office directory. Refer to "Step 12. Add Application Maps and Link to Message Manager" for more information on the relationship between the **Put** location and triggering application maps.

When defining **Get** File Post Offices in Partner Manager, it is not necessary to create new physical directories for triggering files. You can define all **Get** File Post Offices to point to the existing folder *install\_dir\tmgr\_vn.n\mmgr\mail*. The role of the post office and its relationship to physical directory structures is explained in the "StampAndSort Subsystem" documentation.

There is no benefit to defining multiple physical **Get** File Post Office directories. Using the default *install\_dir\tmgr\_vn.n\mmgr\mail* directory simplifies your implementation if you are using this default directory for all File post offices. However, you can customize your implementation according to your needs.

### **stampandsortack map**

The Trading Manager uses the **stampandsortack** map when generating Functional Acknowledgements. The **stampandsortack** map is configured to run and requires no user intervention. The **stampandsortack** map can also generate an alert message to the **stampandsortalert** map if the Message Manager should generate an invalid Functional Acknowledgement (**ack**) triggering name.

## Resend map

The **resend** map works with the Resend feature in Partner Manager to retrieve previously archived data for resending through the Message Manager. The **resend** map has been configured to run correctly and should require no user intervention.

## TRADACOMS subsystem

The **TRADACOMS** subsystem contains two subsystems: **TRADACOMS Inbound** and **TRADACOMS Outbound**.

- The **TRADACOMS Inbound** subsystem provides sorting, validating, and routing functions for data flowing from external trading partners to internal trading partners.
- The **TRADACOMS Outbound** system provides enveloping, validating, and routing functions for data flowing from internal trading partners to external trading partners.

It should be understood that the EDI wizard is not used for managing TRADACOMS standards in the Message Manager system.

## TRADACOMS inbound subsystem

The **TRADACOMS Inbound** subsystem performs the validation and routing functions on inbound TRADACOMS formatted data. The EDI data flows through three primary maps: **TRADACOMS Inbound**, **TRADACOMS\_in\_alert\_data**, and **TRADACOMS\_Traffic\_Inbound**.

### TRADACOMS\_in\_alert\_data map

The **TRADACOMS\_in\_alert\_data** map is a common map component used throughout the Message Manager. The **TRADACOMS\_in\_alert\_data** map is configured to send alert notifications through e-mail or pager whenever data is rejected.

In the **TRADACOMS Inbound** subsystem, the **alert** map is responsible for sending alert notifications when an error condition occurs in the **TRADACOMS Inbound**, map. The rejected data is written to the *install\_dir\tmgr\_vn.n\mmgr\reject* directory and the appropriate form of notification is sent to the user-specified destination.

## TRADACOMS outbound subsystem

The **TRADACOMS Outbound** subsystem performs the enveloping, validating, and routing functions on outbound TRADACOMS formatted data. The EDI data flows through two primary maps: the **alert\_data** map and the **TRADACOMS\_Traffic\_Outbound** map.

### alert\_data map

The **alert\_data** map is a common map component used throughout the Message Manager. The **alert\_data** map is configured to send alert notifications through e-mail or pager whenever data is rejected.

In the **TRADACOMS Outbound** subsystem, the **alert\_data** map is responsible for sending alert notifications when an error condition occurs in the **TRADACOMS\_Outbound** map. The rejected data is written to the *install\_dir\tmgr\_vn.n\mmgr\reject* directory and the appropriate form of notification sent to the user-specified destination.

## Value Added Tax (VAT) reports

The TRADACOMS subsystem (inbound and outbound) will create VAT reports for any invoice type messages. For example: INVOIC, CREDIT, and UTLBIL. The VAT report will automatically be created in the default `c:\install_dir\tmgr_vn.n\mmgr\VAT_Reports\` directory.

If you wish to change the default directory, you can change the location in the **Edit Resource** window in the **Resource Registry**.

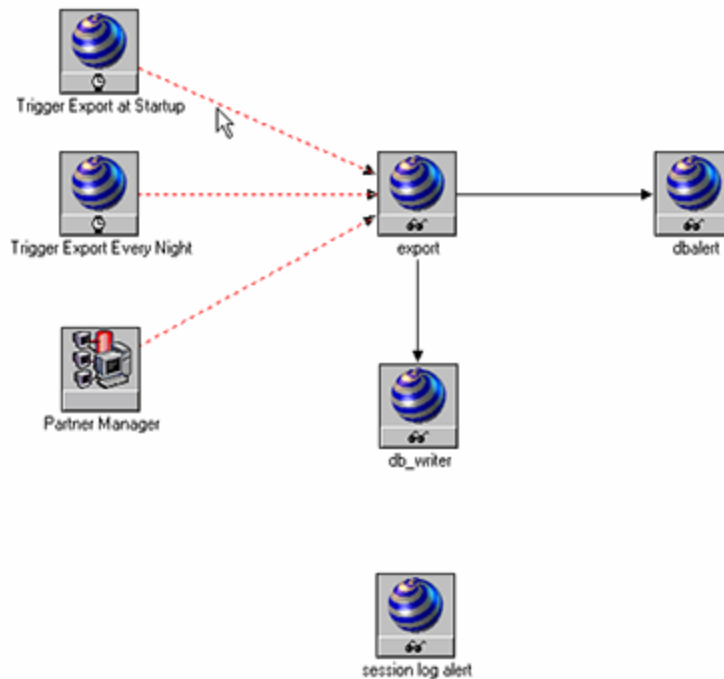
To change the default directory perform the followings steps:

1. Open the **Resource Registry**.
2. Select **Open** from the **File** menu and navigate to the `install_dir\tmgr_vn.n\mmgr\mmgr.mrm`
3. Expand the **mmgr** file.
4. Expand the **Resources** file.
5. Double-click the **mmgr\_trc\_VatReport** file.  
The **Edit Resource** window opens.
6. Edit the **Resource Value** to the desired location.
7. Click **OK**.

## Update subsystem

The **Update** subsystem provides the link between the Partner Manager database and the control files used by the Message Manager. The Partner Manager server database stores all the trading partner information needed to dynamically route, validate, envelope and monitor e-commerce data. This control information needs to be available to the Message Manager during run time to support dynamic routing, validation, and enveloping of data.

The **Update** functions can be scheduled to run as frequently as desired. A system diagram of the **Update** subsystem is shown in the following illustration:



At user-defined intervals, configuration, routing and validation information is extracted from the Partner Manager database and made available to the Message Manager through a set of files with a **.tsv** file name extension. These **.tsv** files contain the configuration, routing, and validation information used as data flows through the Message Manager to dynamically route and transform the data for delivery to its final destination.

### export map

The **export** map is responsible for extracting the data from the Partner Manager database and updating certain state information. The **export** map is triggered to run immediately each time the Launcher is restarted. The run time of the **export** map varies in duration based on the number of trading relationships defined.

As delivered with Message Manager, the **export** map is also triggered to run each night at approximately 12 AM. If you would like to run the **export** map more or less frequently, change the **TimeEvent > Trigger** settings in the Launcher Settings window for the **Trigger Every Night** map of the Update subsystem as shown below. This will allow you to apply immediate changes. Determining the ideal update frequency is a function of how often trading partners are added or updated. If you follow a process that introduces trading partner relationships once a day, the default setting running the **export** map once a day at midnight may be sufficient. If you require more frequent changes, you may want to run the **export** map every 30 minutes.

The **export** map must complete before you can begin processing data. It is recommended that you schedule any user maps with time event triggers to begin a few minutes after the Launcher startup to ensure the availability of the **.tsv** files.

You can trigger the export to run on demand from Partner Manager without cycling the Launcher. Every time the **export** map runs, any changes to trade link information are updated and made available to the **MessageManager** system. The

**MessageManager** controls the transition from the old **.tsv** files to the new ones, so there is not a problem with information changing in the middle of a transaction.

It is not necessary to restart the Launcher when adding new trading relationships for existing version/document combinations. Those trading relationships will be active after the **export** map is run. You must restart the Launcher when adding new version/document combinations, as they require a rebuild of the **sortX12** and **sendX12** maps.

### **dbalert map**

The **dbalert** map monitors the **export** map for error conditions. If an error condition occurs, the **dbalert** map sends an e-mail message with an explanation of the error to the e-mail recipient specified in the first output card. The **dbalert** map also saves a copy of the error information to the *install\_dir\tmgr\_vn.n\mmgr\reject\dberror.log* file.

You must configure the **dbalert** map with the appropriate mail type and recipient ID before running the **MessageManager** system. By default, the **Alert** output card is configured to write the message to *install\_dir\tmgr\_vn.n\mmgr\share\Alert.tmp*. See "Step 4. Modify the E-Mail Alert Settings" .

## **X12 subsystem**

The **X12** subsystem contains two subsystems: **X12 Inbound** and **X12 Outbound**.

- The **X12 Inbound** subsystem provides sorting, validating, and routing functions for data flowing from external trading partners to internal trading partners.
- The **X12 Outbound** system provides enveloping, validating, and routing functions for data flowing from internal trading partners to external trading partners.

## **X12 inbound subsystem**

The **X12 Inbound** subsystem performs the validation and routing functions on inbound ANSI X12 formatted data. The EDI data flows through the **sortx12 map**.

### **sortx12 map**

The **sortx12** map checks the ISA and GS control structures for valid trading partner information.

The **ISA** fields examined include:

- Interchange Sender ID Qualifier
- Interchange Sender ID
- Interchange Receiver ID Qualifier
- Interchange Receiver ID
- Test Indicator

The **GS** fields examined include:

- Functional ID Code
- Application Sender ID
- Application Receiver ID
- Version Release Indicator Code

These values are compared to trade links defined in Partner Manager based on the **Get** post office. If no match is found and an **Unknown Partner** link has not been defined, the data is rejected through the **alert** map.

### **X12 FA inbound map**

The X12 FA Inbound map updates outbound traffic information with the status reflected in inbound Functional Acknowledgements.

### **X12\_In\_Alert\_Data map**

The **X12\_In\_Alert\_Data** map is a common map component used throughout the Message Manager. The **X12\_In\_Alert\_Data** map is configured to send alert notifications through e-mail or pager whenever data is rejected.

In the **X12Inbound** subsystem, the **X12\_In\_Alert\_Data** map is responsible for sending alert notifications when an error condition occurs in the **sortx12** map. The rejected data is written to the *install\_dir\tmgr\_vn.n\mmgr\reject* directory and the appropriate form of notification is sent to the user-specified destination.

## **X12 outbound subsystem**

The **X12 Outbound** subsystem performs the enveloping, validating, and routing functions on outbound ANSI X12 formatted data. The EDI data flows through the **sendx12** map.

### **x12\_out\_alert map**

The **x12\_out\_alert** map is a common map component used throughout the Message Manager. The **x12\_out\_alert** map is configured to send alert notifications through e-mail or pager whenever data is rejected.

In the **X12 Outbound** subsystem, the **x12\_out\_alert** map is responsible for sending alert notifications when an error condition occurs in the **sendx12** map. The rejected data is written to the *install\_dir\tmgr\_vn.n\mmgr\reject* directory and the appropriate form of notification sent to the user-specified destination.

---

## Chapter 6. Support for proprietary data types

These instructions describe how to add proprietary data types to the Trading Manager, which supports EDI standard and proprietary data types. You can add other user-defined data types to the Trading Manager to take advantage of the Message Manager.

---

### Inbound data flow

The Message Manager consists of several subsystems performing specific functions. The Message Manager maps represent a framework for handling e-commerce (EC) data. Any data type may enter the Message Manager through one of the resource adapters. Regardless of direction, the e-commerce data always passes through the **StampAndSort** system first. The **StampAndSort** system assigns a unique **ThreadID** to the e-commerce data object to facilitate multithreaded processing. The **ThreadID** also provides a way to track the e-commerce object through the rest of the system. For information on the Message Manager maps and systems, see the Message Manager subsystems documentation in *Installing and Using Message Manager*.

After the **StampAndSort** subsystem processes the e-commerce data, the subsystem receives it for sorting. **StampAndSort** is based on the default **Ecmail.mtt** type tree and uses the partitioning concept to route e-commerce data dynamically based on its type.

The default **Ecmail.mtt** type tree supports ANSI X12, UN/EDIFACT, TRADACOMS and a Trading Manager proprietary data type. **StampAndSort** rejects any other data type as invalid and passes it into the reject handling system.

You may want to add other user-defined data types to Trading Manager to take advantage of the general EC framework.

---

### Adding proprietary data types

You only need to define enough of the proprietary data structure to make it distinguishable from the other partitioned subtypes.

The figure in the following steps shows an **Ecmail.mtt** type tree that has been modified to include a new proprietary data type named **Test**. The group windows on the right show the component rules required to make the different data types distinguishable.

#### To add support for a new data type

1. Modify the type tree **Ecmail.mtt** to include a new data type **Test** under the partitioned object **M4EC**.
2. Analyze the **Ecmail.mtt** type tree to ensure your new data type is distinguishable.
3. Save the **Ecmail.mtt** type tree.
4. Build the **stampandsortfile** map component in **MessageManager** using the **Integration Flow Designer**.

The Message Manager is now configured to accept proprietary data. For inbound and outbound proprietary data, follow the same rules and naming conventions as you do with EDI data.

## To add application processing for proprietary data

1. Add your own map or system of maps designed to transform the proprietary data and pass it to an appropriate application.
2. Configure the map to trigger on: `c:\install_dir\tmgr_vn.n\mmgr\sortec\*i.Test` (this configuration assumes inbound transmission).

**Note:** Note `*i.Test` represents the actual partition name with `Eemail.mtt`. For instance, if a proprietary data type is added for **NACHA** data formats, the resulting filename from which to trigger would be `*.NACHA`.

Your proprietary data will be processed like any data type in terms of audit trails and alerts. You will be able to track inbound and outbound transmissions in the Reports section of Partner Manager.



---

## Chapter 7. Using alias files in maps

Trading Manager maintains three additional pieces of user data when you define an application partner. This data is contained in the Alias Name, User Defined 1, and User Defined 2 fields and maintained in the Partner Manager database.

You may use this data in maps to reference application partner identification and EDI address information. To use this data:

- Define the alias or user defined fields
- Make the alias available to the application maps
- Choose the alias file to use

---

### Defining alias name and user defined fields

You define alias and user defined fields when you create or edit application trading partners.

For more information on how to specify an alias name and user defined fields for an application partner, see the Partner Manager documentation.

---

### Making the alias data available to application maps

Every time the Message Manager **export** map runs, any changes to trade link information are updated and made available to the **Message Manager** system. It produces four files when it is run; each file contains information about each application partner, including:

- Alias Name
- User Defined 1
- User Defined 2

Each file contains the same data but the columns are ordered differently within the row and the rows are sorted differently within the file:

Description	File Name	Row Sort Order
Alias Information by Partner ID	AliasByP.tsv	By ISA Address (X12)
Alias Information by Alias Name	AliasByA.tsv	By Alias Name
Alias Information by User Defined 1	AliasBy1.tsv	By User Defined 1
Alias Information by User Defined 2	AliasBy2.tsv	By User Defined 2

The alias files are written to the Message Manager . **tmgr\_vn.n \mmgr\share** directory.

The Message Manager does not use these files. They are provided to reference information between your applications' identification methods and the

corresponding EDI addresses. For instance, you might use Alias Name to store your internal customer ID, account number, or client code. Using the appropriate alias files in your application maps, you could move easily between your internal coding systems and your EDI partner addresses.

---

## Choosing the alias file to use

Because the four alias files are sorted differently, you can choose which file most efficiently finds the application coding value or EDI address value, based on your application need and how you have defined the alias and user-defined fields in the Partner Manager.

For example, in an inbound EDI application map, you could use **AliasByP.tsv** (alias by partner ID) to get the appropriate alias name or user defined field for the external trading partner. For an outbound EDI application map where your application data contains the account number for each trading partner, use **AliasBy1.tsv** (alias by User Defined 1, if you have defined the User Defined 1 field to contain your account number).

---

## Alias files type tree (Alias.mtt)

The **Alias.mtt** type tree describes the format of each alias file and is installed in the same directory as the Message Manager source maps and trees.

The **Alias.mtt** type tree contains definitions for each alias file:

Type	Description
AliasByAlias	By Alias Name
AliasByPartner	By ISA Address
AliasByUserDefined1	By each User Defined field
AliasByUserDefined2	By each User Defined field

All four files use the same column definitions; the only difference between each Row definition is the sequence of the columns within the row. The first column corresponds to the sort order of the file.

This type tree can be used in your application maps where you want to use one or more of the alias files.

---

## Creating user type trees

### Overview

The User Type trees are created by the EDI Wizard located in Partner Manager. These type trees are for use in your application maps. If you have both X12 and EDIFACT data, two trees will be generated: **x12user.mtt** and **edifuser.mtt**. These trees are generated in the same directory as the mail trees used by Message Manager. The default location for this is the Message Manager directory: **C:\install\_dir\tmgr\_vn.n\mmgr**. For information on how to use the EDI Wizard, see the Partner Manager documentation.

The EDI Wizard creates the type trees based on the X12 and EDIFACT inbound and outbound trade links defined in the Partner Manager. It automatically creates

a type tree that contains the type definitions for all transaction sets for all versions that the internal trading partners and external trading partners exchange.



---

## Chapter 8. Checklists

The following checklists can be used as site preparation and installation checklists and may be reproduced in any form for use with Trading Manager systems.

Server Installation Checklists:

- Server Installation Checklist for Microsoft SQL Server
- Server Installation Checklist for Oracle Database
- Server Installation Checklist for Sybase
- Server Installation Checklist for IBM DB2 Database

Client Installation Checklists:

- Client Installation Checklist for Microsoft SQL Server
- Client Installation Checklist for Oracle Server
- Client Installation Checklist for Sybase
- Client Installation Checklist for IBM DB2 Server

---

### Server installation checklist for microsoft SQL server

The following checklist identifies the major installation steps required to install and configure Trading Manager for use with a Microsoft SQL Server database. During installation, fill in the specific information in the blanks provided in this list for future reference.

\_\_\_\_\_ System meets minimum requirements

Machine Name \_\_\_\_\_

IP Address \_\_\_\_\_

Modem Brand \_\_\_\_\_

Modem Model \_\_\_\_\_

Modem COM Port \_\_\_\_\_

\_\_\_\_\_ Install Microsoft SQL Server version 6.5, 7.0 (with Microsoft SQL Server Service Pack 3) or Microsoft SQL Server Version 2000 on the server

Database Device Name \_\_\_\_\_

Log Device Name \_\_\_\_\_

Database Name \_\_\_\_\_

Userid \_\_\_\_\_

Password \_\_\_\_\_

\_\_\_\_\_ Install the Launcher

Directory Path \_\_\_\_\_

\_\_\_\_\_ Install Trading Manager Server

Directory Path \_\_\_\_\_

---

## Client installation checklist for Microsoft SQL Server

The following checklist identifies the major steps required to install and configure Trading Manager when connecting to a Microsoft SQL database server. During installation, fill in the specific information in the blanks provided in this list for future reference.

- \_\_\_\_\_ System meets minimum requirements
- \_\_\_\_\_ Microsoft SQL Server database has been constructed on the server
- \_\_\_\_\_ Microsoft SQL Client Services have been installed on each Partner Manager
- \_\_\_\_\_
- \_\_\_\_\_ Check ODBC driver for SQL on each Partner Manager
- \_\_\_\_\_ Install Design Studio
- \_\_\_\_\_ Define ODBC DSN
- \_\_\_\_\_ DSN Directory Path \_\_\_\_\_
- \_\_\_\_\_ Install Message Manager - Message Manager Client (either Application or Design)
- \_\_\_\_\_ Directory Path \_\_\_\_\_

---

## Server installation checklist for Oracle database

The following checklist identifies the major installation steps required to install and configure Trading Manager for an Oracle database server. During installation, fill in the specific information in the blanks provided in this list for future reference.

- \_\_\_\_\_ System meets minimum requirements
- \_\_\_\_\_ Machine Name \_\_\_\_\_
- \_\_\_\_\_ IP Address \_\_\_\_\_
- \_\_\_\_\_ Modem Brand \_\_\_\_\_
- \_\_\_\_\_ Modem Model \_\_\_\_\_
- \_\_\_\_\_ Modem COM Port \_\_\_\_\_
- \_\_\_\_\_ Install Oracle version 7.x, Oracle8.x, or Oracle9.x on the server
- \_\_\_\_\_ User Name \_\_\_\_\_
- \_\_\_\_\_ Password \_\_\_\_\_
- \_\_\_\_\_ Install Launcher
- \_\_\_\_\_ Directory Path \_\_\_\_\_
- \_\_\_\_\_ Install Trading Manager Server
- \_\_\_\_\_ Directory Path \_\_\_\_\_

---

## Client installation checklist for Oracle Server

The following checklist identifies the major steps required to install and configure Trading Manager connecting to an Oracle database server. During installation, fill in the specific information in the blanks provided in this list for future reference.

_____	System meets minimum requirements
_____	Oracle database version 7.x, Oracle8.x, or Oracle9.x installed
_____	Oracle Client Utilities have been installed on each Partner Manager
_____	Install ODBC driver for Oracle on each Partner Manager
_____	Install Design Studio
_____	Define ODBC DSN DSN Directory Path _____
_____	Install Trading Manager Server Directory Path _____

---

## Server installation checklist for Sybase

The following checklist identifies the major installation steps required to install and configure Trading Manager for a Sybase. During installation, fill in the specific information in the blanks provided in this list for future reference.

- \_\_\_\_\_ System meets minimum requirements
  - Machine Name
  - IP Address
  - Modem Brand
  - Modem Model
  - Modem COM Port
- \_\_\_\_\_ Install Sybase on the server
  - User Name
  - Password
- \_\_\_\_\_ Install the Launcher Launcher
  - Directory Path \_\_\_\_\_
- \_\_\_\_\_ Install Trading Manager Server
  - Directory Path \_\_\_\_\_

## Client installation checklist for Sybase

The following checklist identifies the major steps required to install and configure Trading Manager connecting to a Sybase. During installation, fill in the specific information in the blanks provided in this list for future reference.

_____	System meets minimum requirements
_____	Sybase has been constructed on the server
_____	Sybase Client Services have been installed on each Partner Manager
_____	Install ODBC driver for SQL on each Partner Manager
_____	Install Design Studio
_____	Define ODBC DSN DSN Directory Path _____
_____	Install Trading Manager Server Directory Path _____

## Server installation checklist for IBM DB2 database

The following checklist identifies the major installation steps required to install and configure Trading Manager for an IBM DB2 database server. During installation, fill in the specific information in the blanks provided in this list for future reference.

- \_\_\_\_\_ System meets minimum requirements
  - Machine Name \_\_\_\_\_
  - IP Address \_\_\_\_\_
  - Modem Brand \_\_\_\_\_
  - Modem Model \_\_\_\_\_
  - Modem COM Port \_\_\_\_\_
- \_\_\_\_\_ Install IBM DB2 Database version 6.1 or higher on the server
  - User Name \_\_\_\_\_
  - Password \_\_\_\_\_
- \_\_\_\_\_ Install the Launcher
  - Directory Path \_\_\_\_\_
- \_\_\_\_\_ Install Trading Manager Server
  - Directory Path \_\_\_\_\_



---

## Client installation checklist for IBM DB2 Server

The following checklist identifies the major steps required to install and configure Trading Manager connecting to an IBM DB2 database server. During installation, fill in the specific information in the blanks provided in this list for future reference.

_____	System meets minimum requirements
_____	IBM DB2 database version 6.1 or higher installed
_____	Install ODBC driver for DB2 on each Partner Manager
_____	Install Design Studio
_____	Define ODBC DSN DSN Directory Path _____
_____	Install Trading Manager Server Directory Path _____



---

## Chapter 9. Implementing custom X12 versions

This section describes how to implement custom X12 versions such as 004010VICS, 005010RAIL, and/or R2/8.

---

### Overview

Trading Manager uses the **x12mail** Type Tree to validate X12 transactions, and Partner Manager Structure Records to identify segment and data element errors if the transaction fails validation by **x12mail**.

If a custom X12 version is a subset of an ANSI X12 version (for example, 4010VICS is a subset of 4010) Trading Manager will first attempt to use Type Tree and/or structure definitions for the specific subset. If these structure definitions are not found, Trading Manager will use the Type Tree and/or structure definitions for the related ANSI X12 version as long as there also are trade links specified for the functional group ids used in the custom X12 version that are using the ANSI X12 versions. When implementing custom X12 versions, create an **x12mail** Type Tree that includes all of the custom X12 versions to be processed.

You can bypass creating structure records but you should define acknowledgments (997s) to report on the group or transaction level only.

---

### Type trees

For each custom X12 version, a Master Type Tree must be created that follows specific rules so that the EDI Wizard can generate the **x12mail** and **x12user** type trees. These rules are as follows:

- The overall organization of the Type Tree must conform to the organization of the X12 Type Trees contained in the IBM WebSphere Transformation Extender Packs for EDI (See the X12 type trees documentation.)
- Under **Inbound/Outbound Partner Function Group ANSI EDI** the name of the custom group must be **F** followed by the custom version (excluding the leading zeroes). For example, to support 004010VICS the name of the group would be **F4010VICS**. Characters in the version name not supported by the Type Designer should be removed, for example, for **R2/8** the name of the group would be **FR28**.
- The component rule for the GS segment inside the custom group has to have the LEFT function removed and the literal must be the complete version string as it appears in the GS Segment. For example, for **4010VICS** the component rule must read: **VersionReleaseIndustryIDCd Element:\$ = "004010VICS"**
- Under **ANSI EDI** the category that contains the transactions for the custom group must be **V** followed by the custom version (excluding the leading zeroes). For example, to support 004010VICS the name of the category would be **V4010VICS**. Characters in the version name not supported by the Type Designer should be removed, for example, for **R2/8** the name of the category would be **VR28**.

Any changes to the transactions, segments, and/or elements inside the custom category will be picked up by the EDI Wizard and included in the resulting **x12mail** and **x12user** trees.

Versions that contain characters not supported (R2/8) must be manually added to the **x12mail** and **x12user** trees.

---

## Partner Manager

To define tradelinks using custom X12 versions perform the following steps:

1. Start Partner Manager and select **File > Utilities**.  
The **Utilities** Navigator opens.
2. Select **Configuration > Standards**.  
The Standards Maintenance window opens.
3. Select **X12 Versions** from the **X12 Standards** drop-down menu.
4. Enter the custom X12 version in the **Version** field as it shows in the GS Segment (excluding leading zeroes). For example: to support 004010VICS enter **4010VICS**.
5. Click **Save**.
6. Perform one of the following:
  - copy the Transactions and Functional ID's from any other X12 version  
or
  - enter the Transactions and Functional ID's manually (see the Partner Manager documentation for more information).  
If the custom version being implemented is a subset of an ANSI version (for example 4010VICS is a subset of 4010) then you do not need to create structure records. If it is not (for example R2/8), then create structure records and load them into Partner Manager. See "Creating Structure Records" .
7. Create X12 tradelinks that use the custom X12 version.
8. Run the EDI Wizard and point to the Master Type Trees created previously.  
Versions that contain characters not supported (R2/8) must be manually added to the **x12mail** and **x12 user** trees.

---

## Message Manager

Once the EDI Wizard has been executed and an **x12mail** Type Tree created, the **RunMaps** system must be deployed. See "Installing and Using Message Manager."

---

## Creating structure records

To create Partner Manager Structure Records

1. Locate the **StructureValidation.dctl** file, in the **tmgr\_vn.n\pmgr** folder and make a copy of it for the custom X12 version. Ensure that the filename extension is **.dctl**.
2. Select a set of entries that most resemble the transactions under the custom X12 version you are implementing. Delete all other records.
3. In the records selected, change the first 10 characters in each row to represent the custom X12 version as it shows in the GS Segment (excluding leading zeroes). Ensure that the size of the field remains the same, for example: replace 4010\_\_\_\_\_ into 4010VICS\_\_ (where \_ represents spaces).
4. Ensure that columns 11 and 12 contain the string **T1** then save the updated file.
5. Start Partner Manager and select **File > Utilities**.
6. Select **Configuration > Standards** to open the **Standards Maintenance** window.

7. Choose **X12 Structure Validation: Import** from the **X12 Standards** drop-down menu.
8. Click **Find File** to browse to the directory where you placed the file that you created previously and select the file.
9. Click **Begin Import** to complete the process.

Contact Trading Manager Support if more complex changes to the Partner Manager Structure Records for the custom X12 version are required.

---

## TPEC to Trading Manager conversion

### Overview

The Trading Partner E-Commerce (TPEC) to Trading Manager Conversion tool aids in the migration of X12 trading information from TPEC to Trading Manager. The tool automates the transfer of information related to partners and relationships and reduces the amount of tedious and error prone typing that you are required to do. This tool uses as input a file created using the TPEC Partner Extract and creates three files for Trading Manager XML Autoload. It also creates a report with warnings and errors found during the conversion.

After the tool is executed and the information is loaded into Trading Manager, some level of editing/customization will become necessary. The tool does provide for a configuration file that allows you to set parameters specific to your particular conversion.

### TPECAddressBook.xml file

This file contains Partner definitions for Trading Manager. They are grouped into "external partners" (the sending partner of an inbound transaction) and "internal partners" (the receiving partner of an inbound transaction) each one with a unique ISA qualifier and ID combination. It is generated from the Relationship records in TPEC with the partner name from TPEC Partner records, ISA information from the TPEC Interchange records (if present, see the TPEC2CM Configuration File documentation), and GS information from the TPEC Relationship record. The conversion tool will generate as many Trading Manager partners as possible, even if they do not have any TPEC trading specifications defined. The TPEC alias records are not used by the conversion. Please note the following about internal and external partners.

#### Internal partners

- ISA definitions are defaulted since Trading Manager does not validate them.
- Partner names are built from the **ISACompanyPrefix** field in the TPEC2CM Configuration File and the ISA qualifier and ID.
- Application Partner names are built from the **GSCompanyPrefix** field in the TPEC2CM Configuration File and the GS ID.

#### External partners

- ISA definitions are migrated from the TPEC Interchange Records.
- Interchange Control Standard is hard coded to "U" for versions previous to 00402 and to the value specified by **DefaultRepetitionChar** in the TPEC2CM Configuration File for versions 00402 and higher.
- Partner Names are build from either the Corporate ID in the Partner Record or, if not found, from either the TPEC Interchange or TPEC Relationship record as

specified by **DefaultPartnerNameFrom** in the TPEC2CM Configuration File. If there are duplicates the ISA Qualifier and ID are appended to the name.

- Application Partner names are built from the Relationship name. If there are duplicates the GS ID is appended to the name.

## TPECPostOffices.xml file

This file contains Post Office definitions for Trading Manager. They represent logical inputs and outputs of data and there is no equivalent representation in TPEC. Since they are required, they are generated as follows:

- **Get Post Office:** A single post office is defined for all inbound and outbound data going into Trading Manager. Its nickname is hard coded to "EDI" and the path to the directory is defined in the TPEC2CM Configuration File.
- **Outbound Put Post Office:** A single post office is defined for all outbound data going out from Trading Manager to the external partners. Its nickname is hard coded to "EDO" and the path to the directory is defined in the TPEC2CM Configuration File.
- **Inbound Put Post Office:** One inbound put post office is defined for each unique combination of inbound Functional Group and Version/Release. This simplifies the implementation of the user maps need to convert inbound EDI data into the application format. The nickname is the two-character Functional Group ID plus one character representing an index for the Version/Releases found. These post offices are built from the TPEC Inbound Trading Specs. The conversion tool verifies that each Transaction and Version/Release combination is defined in Trading Manager.

## TPECTradelinks.xml file

This file contains Tradelink definitions for Trading Manager. It describes the Functional Groups traded between the partners and has to be loaded last into Trading Manager since it requires CM Partner and Post Office records. It is loosely related to the TPEC Trading Specs records that represent Transactions traded between the Partners. Please note:

### Inbound functional groups:

- Both Test and Production Interchanges will be accepted and routed.
- Functional Acknowledgments will be configured based on the settings in TPEC. 999 Transaction Acknowledgments are not supported by the conversion tool and will be replaced by 997 Transactions.
- Functional Acknowledgments are configured to have the same GS Version/Release that the acknowledging inbound Functional Group have.
- HIPAA transactions are hard coded for Type 1 validation if the Acknowledge option is turned on in TPEC.
- Validation By is defaulted to Transaction.
- Tradelinks for Inbound Functional Acknowledgments are not generated (not required for CM Internal processing) unless the option is selected in the TPEC2CM Configuration File.
- All transactions in a Functional Group will be validated and routed.

### Outbound functional groups

- Outbound Acknowledgements are generated only for those Inbound Tradelinks that have the Acknowledgment option turned on.
- ISA Control Number Scheme is set to "I" (by Interchange).
- GS Control Number Scheme is set to "T" (by Trading Partner).

- ST Control Number Scheme is set to "R" (Relative to the Functional Group).
- Application Format is set to "P" (Partial).
- Validate By is defaulted to "I" (Interchange).
- Last used Control Numbers are not converted from TPEC.

## TPEC2CMConfig.txt file

The purpose of the TPEC2CM Configuration file is to allow the user to define parameters to drive the conversion. There are two types of parameters: site-specific constants and override values for fields in the ISA segment. Please note that the beginning and end of the file should not be changed, but entries and comments can be added in between.

Site-specific constants are required to complete the conversion process: they provide values need to successfully create the XML files for the Partner Manager XML Autoload utility. The following site-specific constants are available:

- ISACompanyPrefix
  - Company Name Prefix for CM internal ISA Partners. Max length allowed is 15 characters.
- GSCompanyPrefix
  - Company Name Prefix for CM internal GS partners. Max length allowed is 15 characters.
- DefaultPartnerNameFrom
  - In TPEC Relationships are supposed to be linked with a Partner definition. In case one is not found the conversion program should default to Interchange (I) or Relationship (R) Records for the CM Partner Name.
- InboundPUTFolder
  - Single folder path for all Inbound PUT Post Offices. Make sure you include the last "\" (or "/" if Unix).
- OutboundPUTFolder
  - Single folder path for the EDO Outbound PUT Post Office. Make sure you include the last "\" (or "/" if Unix).
- GETFolder
  - Single folder path for all inputs into Trading Manager. Should match the StampAndSortFile input directory. Make sure you include the last "\" (or "/" if Unix).
- IgnoreInbound997
  - Inbound 997's are processed internally in Trading Manager. Tradelinks are only required if the user wants to process them into some application of their own. Values are Y/N
- DefaultRepetitionChar
  - The ISA Control Standard is hard coded to "U" for ISA Control versions less than "00402". For "00402" and higher this Repetition Character will be hard coded. Character must be defined in ASCII HEX.

Override values are not required. The XML Autoload will verify that the old values for these parameters are valid in Trading Manager and fail if not. If they are not valid the user has the option of adding the failing values to Trading Manager or (by using these parameters) specifying replacement new values that are valid in Trading Manager. The following override values for fields in the ISA segment are available:

IchgQual:XX|YY

ElementSeparator:XX|XX

SubElementSeparator:XX|YY

SegmentTerminator:XX|YY

IchgCtrlVerNo:XXXXX|YYYYY

In the above values:

- X represents the old TPEC value; Y represents the new CM value.
- Old and New values are fixed length.
- All Old and New values have to be represented in ASCII HEX with the exception of IchgCtrlVerNo that has to be a text string.
- Old SegmentTerminator value is one character long. New value can be up to three characters long represented with a space in between. For example, a terminator of "~<CR><LF>" would be represented as: SegmentTerminator:0A|7E 0D 0A

## Running the TPEC to Trading Manager

### Creating the TPEC Partner Extract file

The conversion tool requires a specific file from TPEC. It can be generated by running the following JCL:

```
//USERIEXT JOB ()  
  
//*****  
  
//** TPJ - DATABASE EXTRACT  
  
//*****  
  
//*  
  
//DBEXT EXEC PGM=TPBDRIVE,REGION=4M,  
// PARM='DATABASE-EXTRACT,PROD'  
//REPORT DD SYSOUT=*  
//TPSVSRT DD SYSOUT=*  
//SYSOUT DD SYSOUT=*  
//SYSPRINT DD SYSOUT=*  
//SYSUDUMP DD SYSOUT=*  
  
//*TEPLIB DD DSN=TSI.TP31.GAI.XA.LOAD,DISP=SHR  
//STEPLIB DD DSN=TSI.TP31.DEV.XA.LOAD,DISP=SHR
```



```

// DD DSN=TSI.TP31.DST.XA.LOAD,DISP=SHR
//TPFORM1 DD DSN=TSI.TP31.D3.DEV.TPFORM1,DISP=SHR
//TPDATA1 DD DSN=TSI.TP31.D3.DEV.TPDATA1,DISP=SHR
//TPFIL11 DD DSN=TSI.TP31.D3.DEV.TPFIL11,DISP=SHR
//TPFIL21 DD DSN=TSI.TP31.D3.DEV.TPFIL21,DISP=SHR
//TPFIL31 DD DSN=TSI.TP31.D3.DEV.TPFIL31,DISP=SHR
//TPSTWK01 DD UNIT=SYSDA,SPACE=(CYL,(10,1))
/* ----- *
/* EBCDIC OUTPUT FILE USED BY THE TPEC2CM CONVERSION TOOL *
/* ----- *
//EXTRACT DD DSN=USER.TP31.EXTRACT,
// DISP=(NEW,CATLG),
// UNIT=SYSDA,
// SPACE=(TRK,(50,50),RLSE)
/* ----- *
/* ASCII OUTPUT FILE NOT USED *
/* ----- *
//DBXTBLE DD DSN=USER.TP31.DBXTBLE,
// DISP=(NEW,CATLG),
// UNIT=SYSDA,
// SPACE=(TRK,(50,50),RLSE)
//CONTROL DD *
EXTRACT PARTNERS
/*

```

Once the job completes FTP the file created by the EXTRACT DD name in ASCII mode and place it in the `C:\install_dir\tmgr_vn.n_beta\platform_support\tpec2cm\` folder. Records are fixed length, make sure no trailing spaces are removed during the FTP.

## Setting up the TPEC2CM conversion map

Before you build the maps that make up this tool please configure/verify:

- The tool uses update.mdq from the mmgr directory where the Message Manager source is installed. It provides access to a Trading Manager database that the tool requires. It needs to be configured to point to the target CM database for this conversion process. To simplify testing it is recommended that you first use an MS Access target database even if the final target database is DB2, Oracle, or any other of the supported databases. You can always migrate data from a CM MS Access database to other platforms or rerun the process pointing to another database.
- The target database should not contain any entries in the Address Book, Post Offices, nor Tradelinks.
- The target database has to have the X12 Structure File loaded. If the TPEC system has HIPAA transactions the HIPAA Structure File should also be loaded into the target database before XML Autoload is executed.
- The tool also uses "cmsvc.dll", "m4ecmgmt.mtt", and "mmgr\_library.mms" from the mmgr directory where the Message Manager source is installed. Output card 1 (ProcessCB) of executable map "ConvertTPEC2CM" has to point to the mmgr folder.

## Running the "ConvertTPEC2CM" executable map

1. Open **tpec2cm.mms** and locate executable map: **ConvertTPEC2CM**
2. Edit input card 1 and point to EXTRACT file FTP'ed from the Mainframe.
3. Build ALL maps.
4. Run the **ConvertTPEC2CM** executable map.

The map will create three files for the XML Autoload:

- **TPECAddressBook.xml**
- **TPECPostOffices.xml**
- **TPECTradelinks.xml**

The map will also create two log files with information about the conversion:

- **TPEC to CM Conversion\_XXXX\_YYYY\_session.xml]**
- **ConvertTPEC2CM.log**

The following files are intermediate files useful for conversion debugging:

- **PartnerReportFile.txt**
- **CleanPartnerReportFile.txt**
- **PostOfficeReportFile.txt**
- **TradelinkReportFile.txt**
- **CleanTradelinkReportFile.txt**

## Running the Partner Manager XML Autoload Utility

The XML files created by the conversion map have to be loaded in the order they are listed above. Please note that you might have to run the conversion process several times before you get a clean run. Therefore is best to make a backup of your target database before loading any data. Perform the following steps for each file:

1. Copy file **tm78.dtd** from **tmgr\_vn.n\pmgr** to the **tpec2cm** folder.
2. Start program **PMAutoload.exe** in the **pmgr** folder.
3. Select the appropriate XML input file using the **Find** button.

4. Configure the destination database to match the one configured in the **tpec2cm.mdq** file.
5. Select **Validate Only; Do not change Database, Allow Add, Detailed Report, and Reject data that exceeds size.**
6. Click **Begin.**

At this point the XML Autoload will validate the input file for format and content using the target database and any error are reported. Usually there are two types of errors:

- ISA field values that are not supported by Trading Manager. Either update the **TPEC2CMConfig.txt** file or, using the Standards option in Partner Manager, add the unsupported values to Trading Manager.
- Duplicate ISA Qual/ID combinations. This can happen when the new value for "IchgQual" in the TPEC2CM Configuration file is already used by some other partner. This can be resolved by either selecting a new value in the configuration file that is not used or changing the ISA information in TPEC for one of the duplicate partners.

Once the files are valid then proceed with the load by selecting "Run Load" in the XML Autoload. The most typical error during this step is:

- Missing structure files in Trading Manager. Either the structure files have not been loaded into the target database of the TPEC system is using non-standard transaction-version/release combinations. The user must either load or create and load structure files indicated by the errors.

---

## Warnings and error messages

Warnings are created to log issues during processing for information purposes only. They do not require attention unless the customer desires a different behavior.

**ID="10001-W">Relationship: GSSENDER Partner Name: 'ABC COMP' is a duplicate. Replaced with: 'ABC COMP ZZISASENDERID'**

In the process of creating external partner names two different ISA Qualifier/ID generated the same Partner Name for Trading Manager. The are made unique by adding the ISA Qual/ID to the name.

**ID="10003-W">Relationship: CONTROL Interchange: BENCH ISA Qualifier: AD is invalid. Replaced with: 19**

The conversion tool found a TPEC ISA Qualifier that matched an old value in the TPEC2CM Configuration file and replaced it with the indicated new value.

**ID="10004-W">Relationship: CONTROL Interchange: BENCH ISA Control Version: 00202 is invalid. Replaced with: 00200**

The conversion tool found a TPEC ISA Control Version that matched an old value in the TPEC2CM Configuration file and replaced it with the indicated new value.

**ID="10005-W">Relationship: BSS Interchange: BSS Sub Element Delimiter: D7 is invalid. Replaced with: 01**

The conversion tool found a TPEC Sub Element Delimiter that matched an old value in the TPEC2CM Configuration file and replaced it with the indicated new value.

**ID="10006-W">Relationship: CONTROL Interchange: BENCH Element Delimiter: D7 is invalid. Replaced with: 02**

The conversion tool found a TPEC Element Delimiter that matched an old value in the TPEC2CM Configuration file and replaced it with the indicated new value.

**ID="10007-W">Relationship: GAPFUNC Interchange: GAPTEST Segment Terminator: B9 is invalid. Replaced with: 0D 0A**

The conversion tool found a TPEC Segment Terminator that matched an old value in the TPEC2CM Configuration file and replaced it with the indicated new value.

**ID="10008-W">Relationship: GSGREG Could not find Partner records for Interchange: ABC. Using ABC**

The conversion tool found a Relationship in TPEC that had an associated Interchange record but could not find a Partner record. Usually happens when relationships are defined but no actual Trading Specs are associated to them. A CM Partner definition is still being created.

**ID="10009-W">Relationship: CONTROL1 Found new GS ID: 'GSRECV'. Created new Application Partner called: 'CONTROL1 GSRECV'**

The conversion tool found multiple Relationship records that contain the same GS ID.

**ID="10010-W">Relationship: KAYBGIN Interchange: KAYBGIN ISA ID: 'FDEX' is a duplicate of an External Partner. Replaced with: '13#FDEX'**

While creating internal partners it found ISA information that matches an already defined external partner. The tool created a dummy ISA ID to allow the load but this should be analyzed.

**ID="10011-W">Relationship: KAYBGIN TradingSpec: 000071 Outbound FA: 999 not supported by this conversion tool. Replaced with 997**

The conversion tool does not support 999 Acknowledgments.

**ID="10012-W">Relationship: HELEN TradingSpec: 000056 FA Relationship: HELENT FA to third party not supported by this conversion tool. Using current Relationship**

The conversion tool does not support acknowledgments to third parties. The tool used the current relationship to allow the load but this should be analyzed.

**ID="20001-C">Relationship: KAYBGIN TradingSpec: 000071 Inbound Transaction: 875 Version/Release: 3020UCS not supported in Trading Manager**

Inbound Transaction-Version/Release not supported. No tradelink will be generated. Users have to add this Transaction-Version/Release to Trading Manager to migrate this Trading Spec.

**ID="20002-C">Relationship: KAYBG TradingSpec: 000166 Outbound Transaction: 875 Version/Release: 3020UCS not supported in Trading Manager**

Outbound Transaction-Version/Release not supported. No tradelink will be generated. Users have to add this Transaction-Version/Release to Trading Manager to migrate this Trading Spec.

**ID="20003-C">Relationship: ERISCO TradingSpec: 000045 Outbound FA: 997 Version/Release: 3032ERISCO not supported in Trading Manager**

Outbound Transaction-Version/Release not supported. No tradelink will be generated. Users have to add this Transaction-Version/Release to Trading Manager to migrate this Trading Spec.



---

## 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 106-0032, 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  
577 Airport Blvd., Suite 800  
Burlingame, CA 94010  
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 measurements 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 illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.



---

## Programming interface information

Programming interface information, if provided, is intended to help you create application software using this program.

General-use programming interfaces allow you to write application software that obtain the services of this program's tools.

However, this information may also contain diagnosis, modification, and tuning information. Diagnosis, modification and tuning information is provided to help you debug your application software.

**Warning:** Do not use this diagnosis, modification, and tuning information as a programming interface because it is subject to change.

---

## Trademarks and service marks

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

AIX  
AIX 5L  
AS/400  
Ascential  
Ascential DataStage  
Ascential Enterprise Integration Suite  
Ascential QualityStage  
Ascential RTI  
Ascential Software  
Ascential  
CICS  
DataStage  
DB2  
DB2 Universal Database  
developerWorks  
Footprint  
Hiperspace  
IBM  
the IBM logo  
ibm.com  
IMS  
Informix  
Lotus  
Lotus Notes  
MQSeries  
MVS  
OS/390  
OS/400  
Passport Advantage  
Redbooks  
RISC System/6000  
Roma  
S/390  
System z  
Trading Partner  
Tivoli

WebSphere  
z/Architecture  
z/OS  
zSeries

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

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

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

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

This product includes software developed by the Eclipse Project (<http://www.eclipse.org/>).



IBM WebSphere Transformation Extender Trading Manager, Version 7.8

---

# Index

## A

Access database 10  
alert\_data map 36  
alias files 43  
    Alias.mtt 44  
archive directory 31  
Archive subsystem 29

## C

configuring  
    Message Manager 18  
    Partner Manager 15  
    Trading Manager 9  
custom X12 versions  
    Message Manager 54  
    overview 53  
    partner manager 54  
    structure records 54  
    type trees 53

## D

Data Source Name  
    defining 6, 15  
Database  
    defining ODBC Data Source Name (DSN) 23  
    IBM DB2 12  
    Microsoft SQL Server 10  
    ODBC vendor links 7  
    Oracle 11  
    query file 22  
    using an Access database 10  
dbalert map 39  
deploy facility 25  
deployment  
    tasks 18  
Design Studio 7

## E

e-mail 24  
    alert settings 21  
Email.mtt 24  
EDI  
    documents 3  
    Integration Package 2  
    Wizard  
        running 24  
EDIFACT  
    subsystem 29  
        inbound 29  
        outbound 30  
edifact\_in\_alert  
    alert 30  
edifact\_in\_alert map 30  
email  
    service 3  
export map 38

## F

FTP 24

## G

Get  
    File Post Office 35

## I

IBM DB2  
    database 12  
    enhancements 13  
import  
    X12 structure validation 15  
inbound  
    archiving 31  
    subsystems  
        EDIFACT 29, 36  
        TRADACOMS 36  
        X12 39  
installation 4, 7  
    checklists 47  
        IBM DB2 database 50  
        IBM DB2 Server 51  
        Microsoft SQL Server 47, 48  
        Oracle database 48  
        Oracle Server 49  
        Sybase 49, 50  
    Launcher 17  
    Message Manager 17  
    Trading Manager  
        components 5  
installationMessage Manager  
    installing 17  
    Message Manager 17  
Integration Flow Designer  
    defined 1  
    deploy facility 25

## L

Launcher  
    defined 1  
    installation 4  
    management tools 27  
    modifying 22  
    on UNIX 3  
    Service  
        starting 27  
links  
    adding 24  
Lotus  
    CCMail 3  
    Notes 3

## M

management tools 27

- maps
  - adding 24
  - alert\_data 36
  - dbalert 39
  - export 38
  - resend 36
  - Send Problem Data Alerts 30
  - sortx12 39
  - stampandsortack 35
  - stampandsortEmail 32
  - stampandsortfile 35
  - stampandsortFTP 32
  - stampandsortVAN 32
  - TRADACOMS\_in\_alert\_data 36
  - X12\_In\_Alert\_Data 40
  - x12\_out\_alert 40
- Message Manager
  - adding maps and links 24
  - configuring 18
  - custom X12 versions 54
  - defined 1
  - deploying on UNIX 20
  - deployment tasks 18
  - installing 17
  - modify settings 19
  - starting 18
  - subsystems 28
- Microsoft
  - Exchange 3
  - Mail 3
  - SQL Server 10
- msg\_mgr.msdl
  - modifying 19

## O

- ODBC
  - driver 6
  - resolving difficulties 6
  - setup 6
  - vendor links 7
- Oracle
  - database 11
  - SQL scripts 11
- outbound
  - subsystems
    - EDIFACT 30
    - TRADACOMS 36
    - X12 40

## P

- Partner Manager
  - configuration 15
  - defined 1
  - system requirements 3
- partners
  - defining 23
- paths
  - design-time 19
  - run-time 19
- post office
  - defining 23
- proprietary data types 41
- Put
  - File Post Office 35

## R

- resend map 36
- Resource Adapters 1

## S

- Send Problem Data Alerts map 30
- SMTP/POP3 3
- sortx12 map 39
- SQL scripts 9
  - Oracle 11
  - running 11
- StampAndSort
  - subsystem 30
- stampandsortack map 35
- stampandsortEmail map 32
- stampandsortfile map 35
- stampandsortFTP map 32
  - copying 34
- stampandsortVAN map 32
- start
  - Launcher Service 27
  - Message Manager 18
- startup parameters
  - Launcher 22
- structure records
  - in custom X12 versions 54
- structure validation
  - importing X12 15
- subsystems 28
  - Archive 29
  - EDIFACT 29
    - inbound 29, 36
    - outbound 30
  - StampAndSort 30
  - TRADACOMS 36
    - inbound 36
    - outbound 36
  - Update 37
  - X12 39
    - inbound 39
    - outbound 40
- Sybase 12
- system requirements
  - Partner Manager 3
  - UNIX 3

## T

- ThreadID
  - Date Received 31
  - definition of 31
  - Direction 31
  - Get Post Office Nickname 31
  - interchange Index Number 31
  - Mailbag Number 31
  - naming conventions 31
  - Origin 31
  - Put Post Office nickname 31
- TRADACOMS
  - subsystem 36
    - inbound 36
    - outbound 36
  - VAT reports 37
- TRADACOMS\_in\_alert\_data map 36

- trade links
  - defining 23
- Trading Manager
  - configuring 9
  - defined 1
  - installation checklists 47
- TX Launcher
  - installation 17
- type trees
  - Alias.mtt 44
  - Email.mtt 41
  - x12user.mtt 44

## U

- UNIX
  - deploying Message Manager 20
  - system requirements 3
- Update subsystem 37

## V

- VAN 24
- VAT reports 37

## X

- X12
  - subsystem 39
    - inbound 39
    - outbound 40
- X12\_In\_Alert\_Data map 40
- x12\_out\_alert map 40
- x12user.mtt 44







Printed in USA