The EDM Manager™

Operations Guide

Legal Information

Trademarks

Enterprise Desktop Manager (EDM), EDM Administrator, EDM Agent, EDM Client, EDM Manager, EDM Stager, and EDM Packager are trademarks of Novadigm, Inc. Other brand names and product names are trademarks or registered trademarks of their respective companies.

No investigation has been made of common-law trademark rights in any word. Unless otherwise noted, all names of companies, products, street addresses, and persons contained herein are part of completely fictitious scenarios, data, or other information, and are intended solely to document the use of Novadigm, Inc. software products.

The Novadigm Enterprise Desktop Manager (EDM) enables you to manage and distribute software throughout your organization. It does not provide you with the right to make copies of software that you have obtained from third parties. You should make sure that you have obtained the right to make and use the number of copies of each software program which you distribute using EDM.

Confidentiality Statement

These materials contain the confidential, proprietary information of Novadigm, Inc., and are for the sole use of the party to which they are provided and solely for use with the Novadigm software with which they are provided, and as may be expressly agreed upon between that party and Novadigm, Inc. Any other dissemination, distribution, copying or use of the information disclosed hereunder is strictly prohibited.

Restricted Rights

The software and accompanying documentation are provided with "Restricted Rights." Use, duplication, or disclosure by the US Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFAR §252.227-7013; in subparagraphs (c) (1) and (2) of FAR §52.227-19, Commercial Computer Software-Restricted Rights; or FAR §52.227.14, Rights in General Data, Alternative III, as applicable. Contractor/Manufacturer is Novadigm, Inc., One International Blvd., Suite 200, Mahwah, NJ, 07495.

Notices

The information contained in this document is subject to change without notice, and does not represent a commitment by NOVADIGM, INC. The software described herein is furnished under licensed agreement and/or nondisclosure agreement. NOVADIGM, INC. software can only be used, copied, or transmitted in accordance with the terms of the license agreement.

No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including but not limited to photocopying, scanning, or information storage and retrieval systems, for any purpose other than for the explicit use by the licensed organization and/or individual person, without the express written permission of NOVADIGM, INC.

Printed in the United States of America

Preface

About This Book

Welcome to EDM. NOVADIGM's Enterprise Desktop Manager (EDM) is a powerful systems management platform that enables you to effectively manage the vast number of enterprise resources that must be supported in today's complex multi-vendor, multi-protocol, distributed computing environments.

This section provides you with an overview of this operations guide, so you can quickly turn to the information you need. The following chapter summaries should help you to quickly and easily locate the information you need.

If You Want To:	Turn To:	To Learn More About:
Install the EDM Manager	Chapter 1	The system requirements and information you will need before you can begin the installation process.
Operate the EDM Manager	Chapter 2	The correct procedures to use when starting and stopping the EDM Manager.
Tune the EDM Manager	Chapter 3	How to tune the EDM Manager for Windows NT by specifying values in the EDMPROF.DAT file.
Maintain the EDM Manager	Chapter 4	How to work with database utilities to maintain the EDM Manager.
Use the EDM Manager log, logging facility, and its related messages	Chapter 5	Interpreting the EDM Manager log file so you can use it in conjunction with the logging facility, and its related messages.
Learn about EDM Manager messages	Chapter 6	The complete list of EDM Manager messages that are generated by the EDM logging facility. You will also learn about interpreting EDM Manager messages, and any system actions that may be pertinent.
Use Windows NT security with your EDM Manager.	Chapter 7	How to define EDM Administrator IDs and passwords to Windows NT security, configure EDM to call the EDMSIGNR method, and define Administrator and Password Information to EDM.
Learn about the EDM Manager Console	Chapter 8	How to monitor system functions, view and adjust task settings and parameters, submit REXX requests, perform NOTIFYs, and shut down the EDM Manager.
Use the EDM Performance Monitor	Chapter 9	The variety and scope of the operation information presented by the EDDM Performance Monitor and how to change display aspects.
Learn about the EDM Manager Add-on features for Version 4.0	Chapter 10	The additional features that can be used with the EDM Manager. These features include SMTP mail support and Client error message retrieval.
Use Manager Methods	Appendix A	The numerous EDM Manager methods. Included for each method is the usage, the required parameters, and return codes.
Export Data using the EDMMSQLP Method	Appendix B	How to specify and extract data from the EDM database and store it in your own SQL database table.

Table of Contents

1 Installing the EDM Manager for Windows NT

This chapter provides you with step-by-step directions for installing the Version 4.0 EDM Enterprise Manager for Windows NT.

This chapter also provides you with information on how to run the Manager as either a non-service or an NT service.

System Requirements

To install and operate the EDM Enterprise Manager for Windows NT at your site, the following system requirements apply:

- Windows NT Server (3.51 or 4.0).
- Pentium processor 120 MHz or higher.
- 16MB of memory.
- While 16MB of memory will allow you to perform the Manager installation, we recommend additional memory to optimize the installation process and subsequent operations.
- NT File System (NTFS) for the EDM database.
- The required amount of available hard disk drive space, as shown in the table below.

Hard Disk Drive Space Requirements

Component	Space
EDM Enterprise Manager Note that this space requirement does not include estimates for the applications you plan to manage by using EDM.	20MB
EDM logs	50MB (recommended)

Note: Before you install the Manager for Windows NT, ensure that the user's account is included in the Administrator's group. If you perform the installation without belonging to the Administrator's group, the Manager will not be installed properly.

- We strongly recommend that you apply the latest maintenance release to your system.
- Minimum virtual memory: 150MB.

You can tune the virtual memory parameter to enhance performance. We recommend a virtual memory setting of at least 200MB.

Note: When you set the Virtual Memory parameters, you must have an Initial and Maximum value in each list box, respectively. Note that Windows NT Version 3.51 will only read the Initial value, while Windows NT Version 4.0 will honor both the Initial and Maximum values.

Note: You can also tune the Tasking parameter, which determines how the NT system runs the Manager: as a foreground task, background task, or both. For more details, see *"Tuning Virtual Memory and Tasking Parameters."* below.

Tuning Virtual Memory and Tasking Parameters

Tuning the virtual memory parameters before installing the EDM Enterprise Manager will facilitate the installation process by ensuring enough paging file space is available to complete the installation program and to run the Manager. Once you have set the space parameters, tune the Tasking parameters to ensure that the Manager runs properly.

Note: If you are installing the Manager using Windows NT 3.51, follow the steps below to tune the Virtual Memory and Tasking parameters.

If you are installing the Manager using Windows NT 4.0, turn to page 227 to tune these parameters, follow the steps to tune the Virtual Memory and Tasking Parameters also in this chapter.

Warning: If you are reinstalling the Manager, we strongly recommend that you shut it down before making these changes to the Windows NT system, and before the reinstallation process.

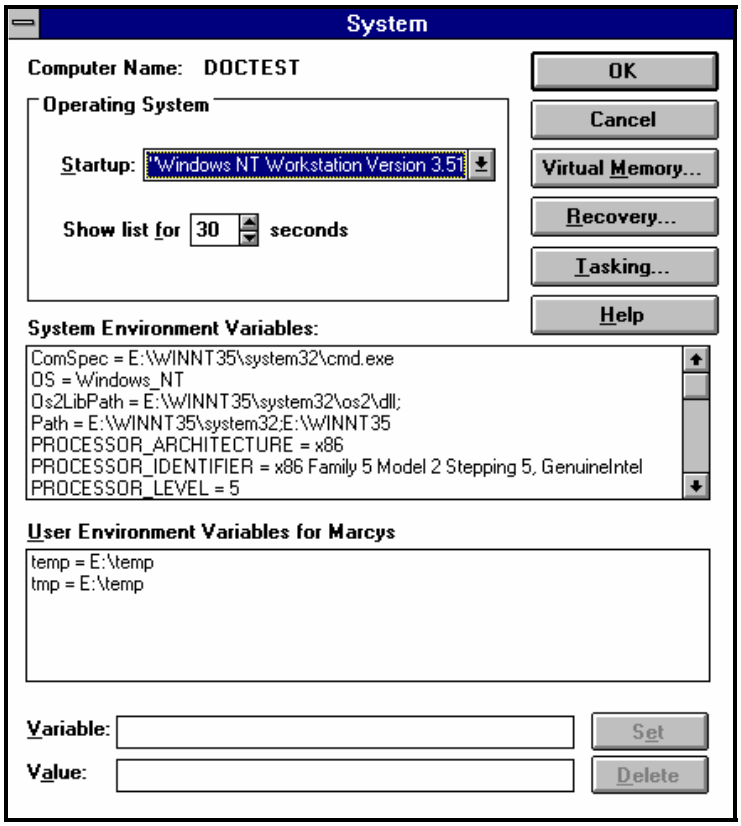
For more information, see to "*Chapter 2: Starting and Stopping the EDM Manager for Windows NT*" in Chapter 2.

Windows NT 3.51

➢ **To Tune the Virtual Memory and Tasking Parameters:**

1 To specify the virtual memory setting, choose the **System** icon from the **Control Panel** folder.

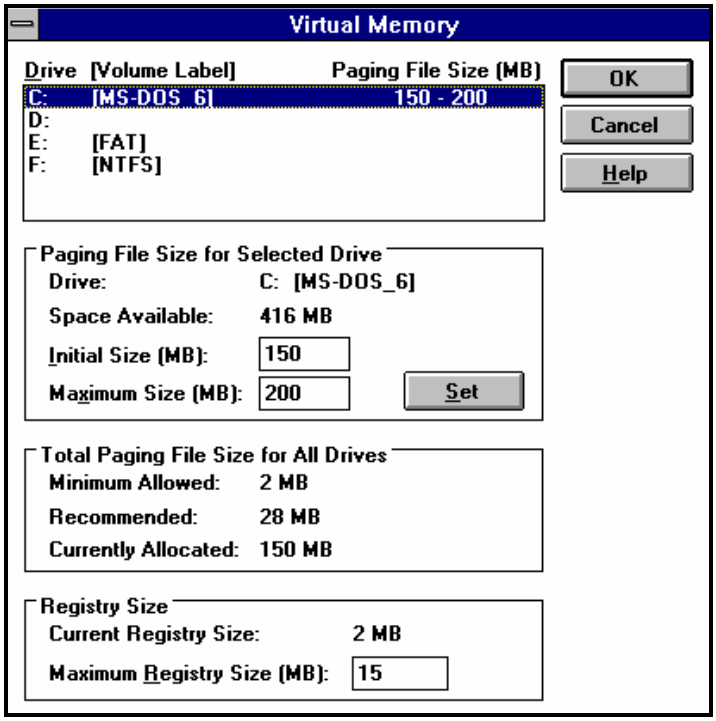
The System dialog box is displayed:.



2 The **Startup** drop down box displays the operating system you are using (for example, Windows NT Server Version 3.51). This Also, this screen also displays the System and User environment variables.

Choose **Virtual Memory**... to set the Paging File Size.

The **Virtual Memory** dialog box is displayed:



3 To view the Paging File Size for a specific drive, highlight that drive in the Drive list box, and review the information in the Paging File Size for Selected Drive group box.

The Initial Size (MB) text box must have a minimum value of 150MB. The recommended value is 200MB or more.

Note: While you can specify a value for the Initial and Maximum items, the Windows NT 3.51 will only read the Initial Size (MB) value.

To change the Initial and/or Maximum paging file size, highlight the value in the appropriate text box, and then type in the desired value.

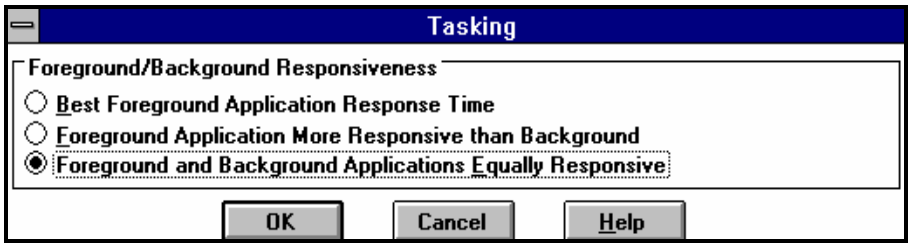
4 Choose **Set** in the **Paging File Size for Selected Drive** box, and then choose **OK**.

The **System** dialog box is displayed again.

Now that you have set the Virtual Memory parameters, tune the Tasking parameter, which determines how the NT system runs the Manager: as a foreground task, background task, or both.

5 In the **System** dialog box, choose **Tasking...**

The **Tasking** dialog box is displayed.:



6 Select the radio button for the **Foreground and Background Applications Equally Responsive** option to ensure that the Manager will run in the foreground and background, equally. Then, choose **OK**.

The **System** dialog box is displayed again.

7 Choose **OK**.

You are prompted to reboot your system for the changes to take effect.

After you reboot your system, you can continue installing the Manager.

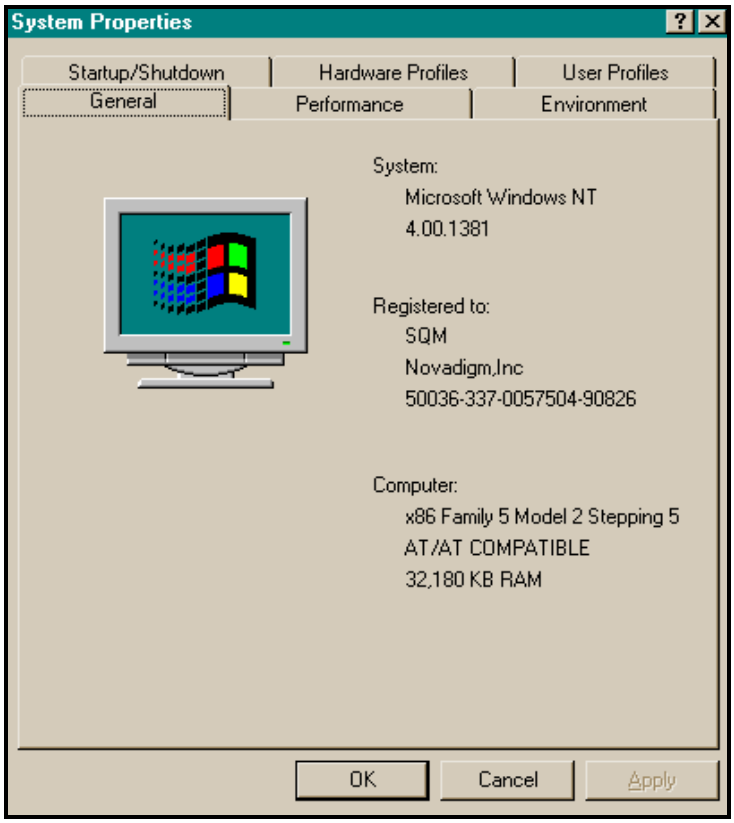
For general NT performance guidelines, refer to The Windows NT Resource Kit.

Windows NT 4.0

➢ **To Tune the Virtual Memory and Tasking Parameters:**

1 To set Virtual Memory and Application Performance tasking parameters, open the Control Panel in the Start menu. Then, select the System icon.

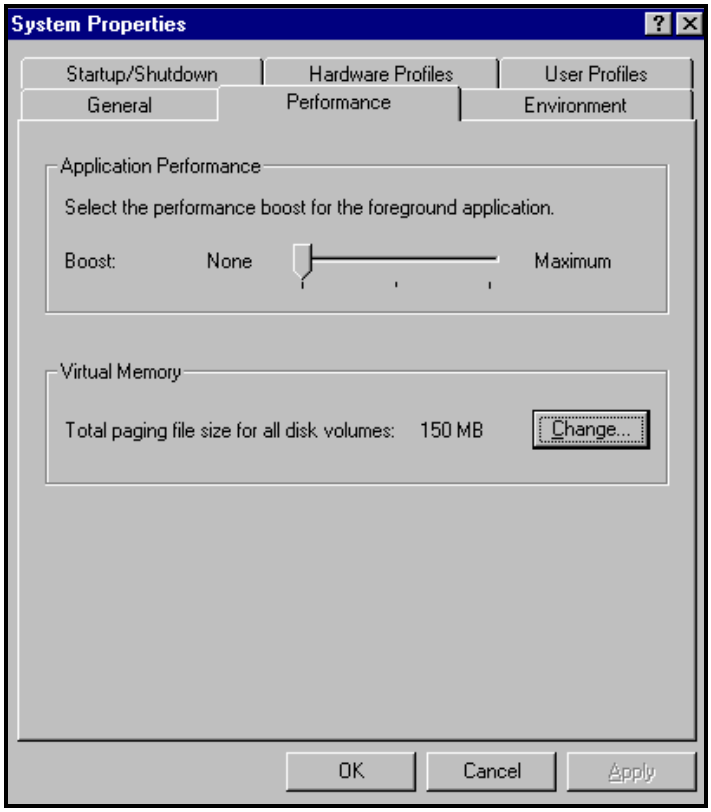
The **System Properties** dialog box is displayed:.



You will use the **System Properties** dialog box to tune the virtual memory and application performance tasking parameters.

2 Choose the **Performance** tab to set the Application Performance tasking parameters and the paging file size in virtual memory.

The following dialog box is displayed.

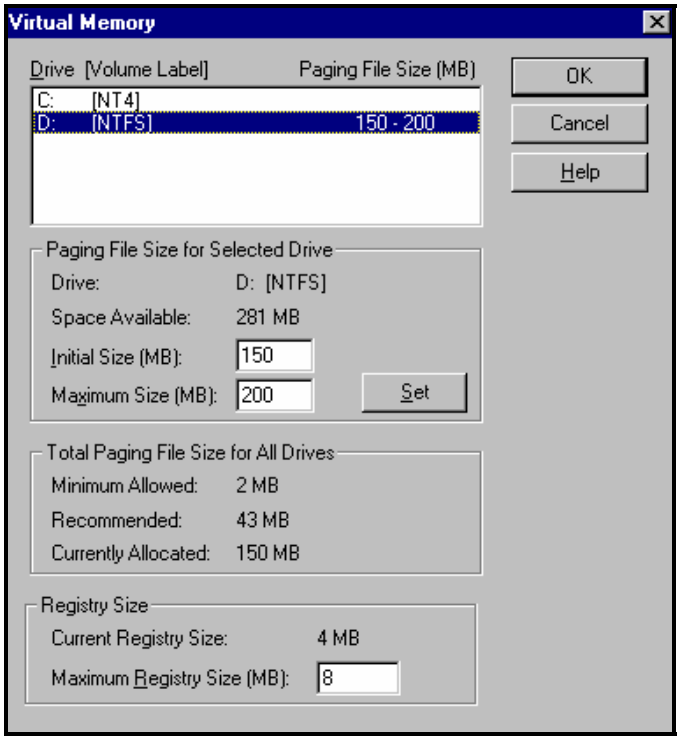


The **Application Performance** box lets you specify whether the NT system will run the Manager in the foreground, background, or both. In the **Virtual Memory** box the paging file size is displayed.

3 Set the **Boost** slide bar to **None** to enable the Manager to run in the foreground and background, equally.

4 Then, set the paging file size for virtual memory by choosing **Change...**

The **Virtual Memory** dialog box is displayed.



5 The Drive box displays information about each drive, including the Volume Label, and the Paging File Size.

To view the Paging File Size for a specific drive, highlight that drive in the **Drive** box and then review the information in the **Paging File Size for Selected Drive** box.

Note: The Initial Size text box must have a minimum value of 150MB. The recommended value is 200MB or more.

To change the Initial and/or Maximum paging file size, highlight the value in the appropriate list box and then type in the desired value.

6 Choose Set in the Paging File Size for Selected Drive box.

Now review the information in the **Drive** box. Ensure that the drive and the paging file size you specified is correct. Then, choose OK.

You have now finished tuning the Virtual Memory and Application Performance tasking parameters.

After you reboot your system, you can continue installing the Manager.

For general NT performance guidelines, refer to The Windows NT Resource Kit.

About the Installation Program

The EDM Enterprise Manager for Windows NT Installation Program comes to you on CD-ROM. Before getting started, you should note the following about the installation program:

- Before you start installing the Manager, close all active applications, exit Windows, and then restart Windows.
- If you are re-installing the Manager on top of a previous version, additional panels will be displayed during the installation.

  Note: When you install this version of the Manager, the program will automatically configure your database to the expanded format, which is required for new Version 4.0 functionality to take effect.

- If you have already installed the Manager on your computer, the installation program will prompt you to specify how you want to install the database files: overwrite the existing database, backup the existing database, or use the existing database.
- When you install the Manager for Windows NT installation program, you will see Novadigm's default selections displayed. Of course, you can override these defaults as you make your selections.
- You can run the Manager as a non-service or as an NT service.

  Warning: Do not simultaneously run the Manager as a non-service and an NT service. If you run both processes at the same time, you may get unexpected results.

- If you want to change your NT configured startup process, you must first reboot your system. Then, configure the Manager for the application you want to use.
- If you want to run the EDM Manager as a non-service, you must choose the EDM Manager Startup icon from the Manager program group.

  The term "non-service" is used to refer to running the Manager startup process manually. There are no NT configuration options available for this application.

  For additional details, see "*Running the EDM Manager for Windows NT as a Non-Servicee,"* later in this chapteron page 37in this chapter.

- If you want to run the Manager as an NT service, you can configure the Manager startup process.

  This service gives you the opportunity to run the Manager manually, or automatically upon rebooting your system, as well as control the starting and stopping of a service.

  In the Windows NT operating system, the term "service" refers to the Manager that interacts with the Service Control Manager. This NT component maintains the Event Log and the Services panel. Also, once the service is running, you may pause the service at any time during that session, and then continue at the point where you paused.

  For additional details, see *"Running the EDM Manager for Windows as an NT Service,"* later in this chapter in this chapteron page 39.

For details on configuring, starting, and stopping an NT service, see "*Chapter 2: Starting and Stopping the EDM Manager for Windows NT NT."* in this chapter.

Navigating the EDM Manager for Windows NT Installation Program

During the installation process, a series of dialog boxes is displayed that allow you to select installation details.

The following helpful hints describe how to get around in the selection dialog boxes:

- To continue to the next dialog box, choose **Next >**.
- To return to the previous dialog box, choose **<** Back.
- To enter text, select the appropriate text box and type the specified information.
- To delete existing text in a text box, simply type over it. Or, use the **Delete** key to erase unwanted characters.
- To specify a different drive or directory, choose **Browse**.
- To exit the installation program from within the dialog box, choose **Cancel**.

> **Warning**: If you are reinstalling the Manager, you should shut it down before starting the reinstallation process. For more information, see "*Chapter 2: Starting and Stopping the EDM Manager for Windows NT*." in Chapter 2.

Installing the EDM Manager for Windows NT

The following information explains how to install the EDM Enterprise Manager.

Warning: Before you install the Manager for Windows NT, check to ensure that the user's account is included in the Administrator's group.

Use the Group Memberships dialog box to establish or change the membership of the user's account.

If you perform the installation without belonging to the Administrator's group, the Manager will not be installed properly.

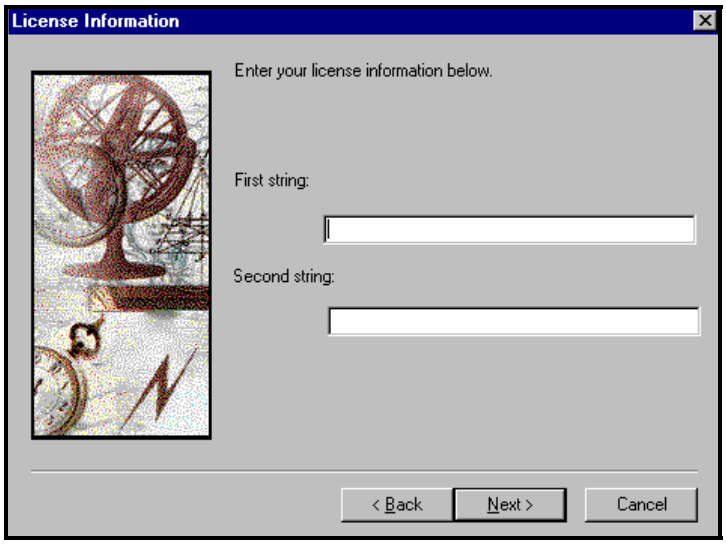
Note: Before you start installing the Manager for Windows, close all active applications, exit Windows, and then restart Windows.

➢ **Installing the EDM Enterprise Manager for Windows NT:**

1. Insert the installation CD-ROM into the CD-ROM drive.
2. Choose the **Run...** command.

   Type `D:\MANAGER\DISK1\SETUP.EXE`, and then choose **OK**. (If your CD-ROM has another drive letter, specify it.)
3. Choose **Next >**.

   The following dialog box is displayed.



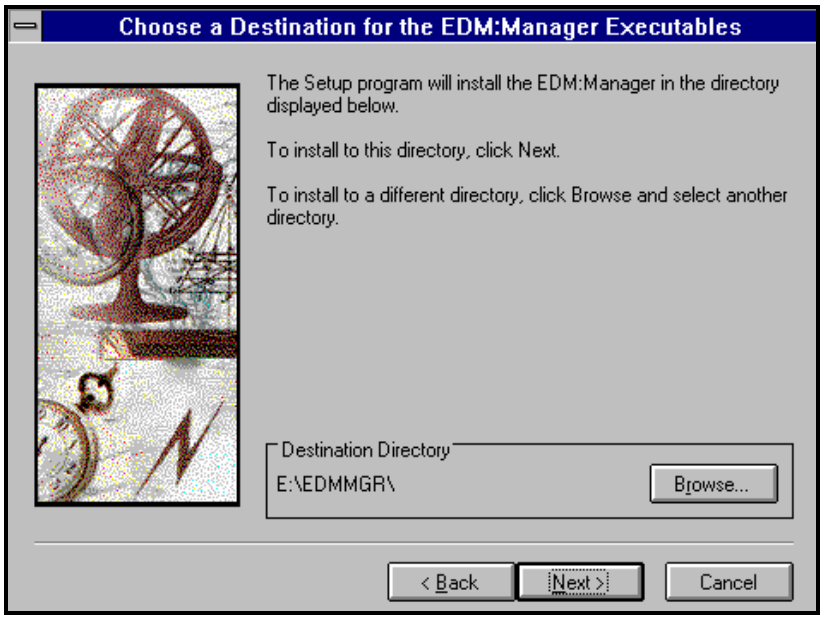
4. Enter your license information., including spaces, in the **First String** text box. The **Second String** text box is not being used at this time.

   In order to complete the EDM Manager installation it is necessary for you to have a valid license string.

   Refer to the cover of your installation CD-ROM jewel case for information on your USERID and password for Novadigm's FTP site. The license string is located in the license.txt file. Copy the two license strings and paste them into the appropriate text box by using **Shift+-Insert** on your keyboard.

   Note: The license string uses the numeral 0. The letter O is not used.

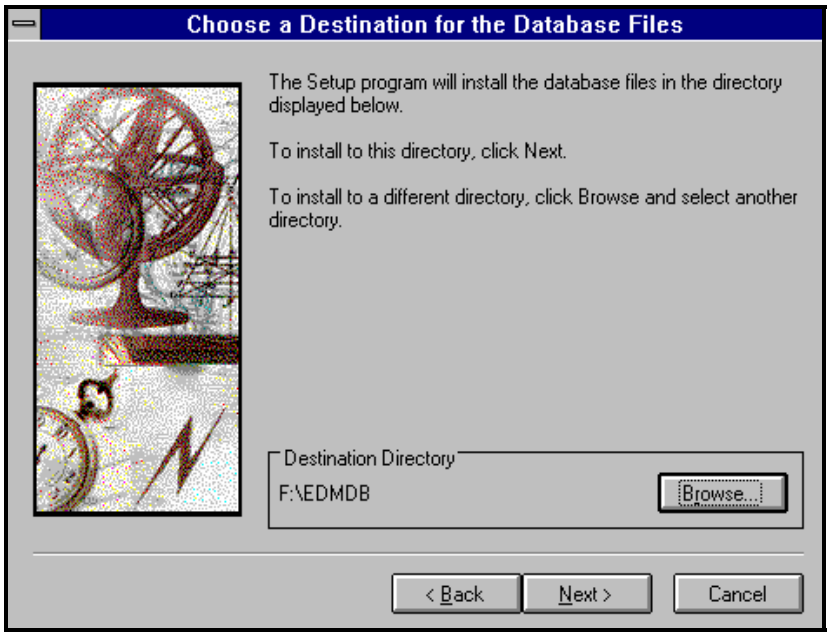
The following dialog box is displayed:



5 To install the Manager to the directory that is already specified, choose **Next .>.** Or, choose **Browse** to specify another destination directory for the Manager files.

If the Manager was previously installed into the displayed directory, you will be prompted to confirm your selection.

The following dialog box is displayed.:



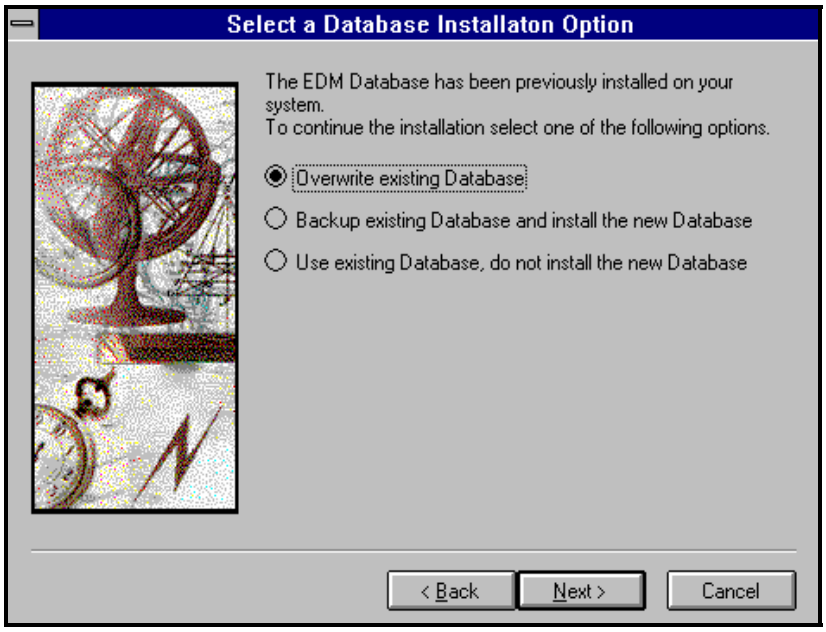
6 Specify a destination for the database files, for example `F:\EDMDB`.

To install the database files in the default directory, choose **Next >**. Or, choose **Browse** to specify another directory for the files.

You must install the database files into an NTFS directory. Note that the Manager and the database files can be stored in the same directory and/or drive.

Note: If you are installing the Manager on your desktop for the first time, or installing a new database directory, turn to step **8**.

If you have previously installed the Manager on your desktop, and you want to install this version of Windows NT using the same database directory, the following dialob dialog box is displayed:.



7 Specify one of the available options. Then, choose **Next >**.

The following table lists and describes these three configuration options.

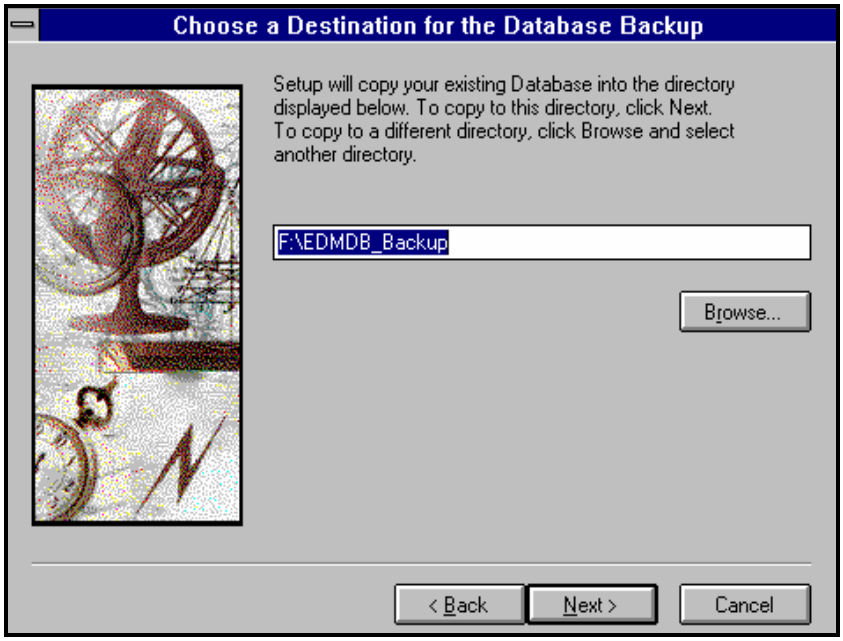
Note: All three options will have the expanded database format that comes with this version of the Manager for Windows NT.

The Configuration Option List

Configuration Option	Result
overwrite existing database	Installs the “as shipped” EDM database. Note: If you have customized the existing database, those changes will be discarded when the new database is installed.
backup existing database	Installs the “as shipped” EDM database. Note: If you have customized your existing database, those changes will be saved in the backup only. The backup is placed in the drive and directory you specify.
use existing database	Converts the EDM database to the appropriate *format* while retaining your existing database *contents*. This version is in the expanded format.

Depending on the option you selected, you will need to specify additional information.

- If you selected the backup option, turn to step **68.**



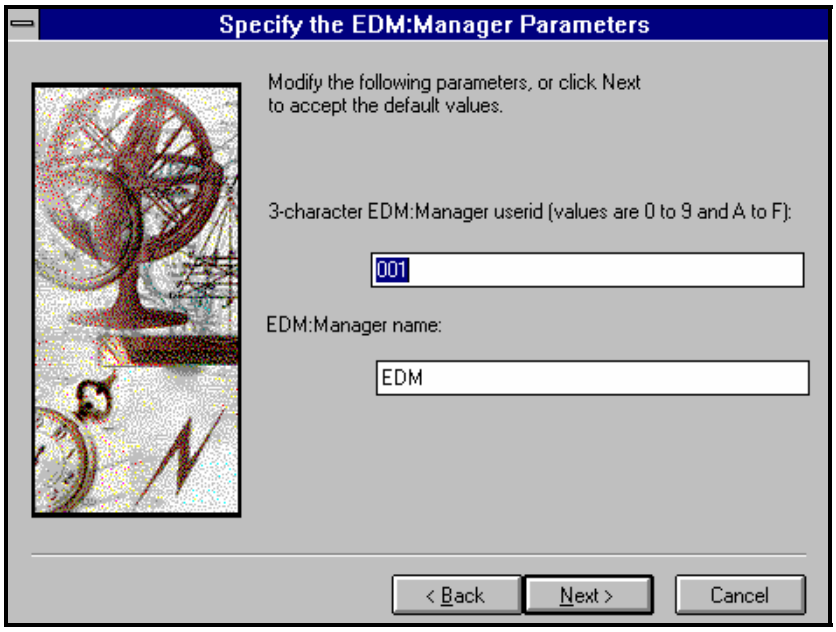
- If you opted to overwrite the existing database or perform the installation using the existing database, turn to step **79**.

8 The installation program creates a backup copy of your database and automatically specifies a drive path based on the name of the database directory that you specified earlier, for example:

`F:\EDMDB_Backup.`

To accept the value displayed, choose **Next >**. To specify a different directory name, choose **Browse**.

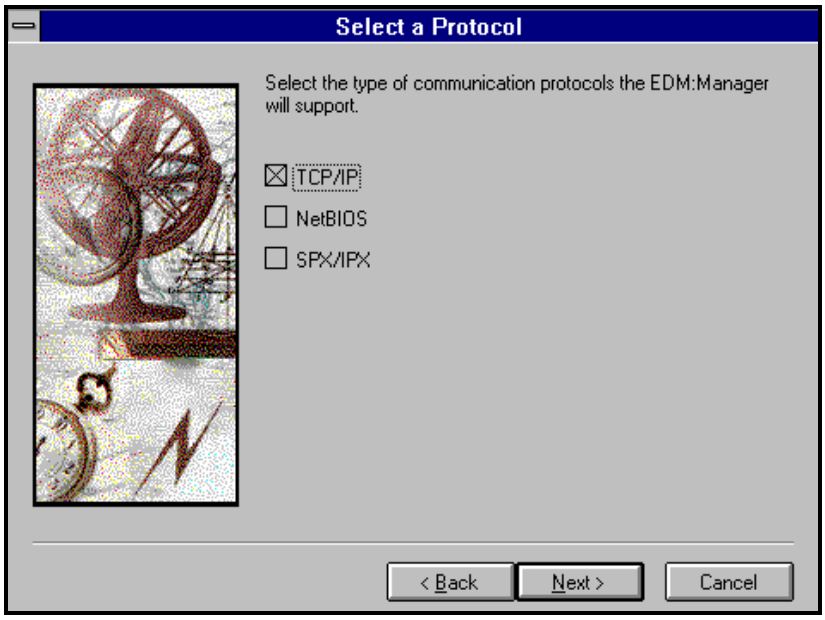
The following dialog box is displayed.



9 To accept the value shown, choose **Next >**. Or, specify a valid 3-character Manager userid. Note that the Manager ID is used as the prefix in the Manager log file.

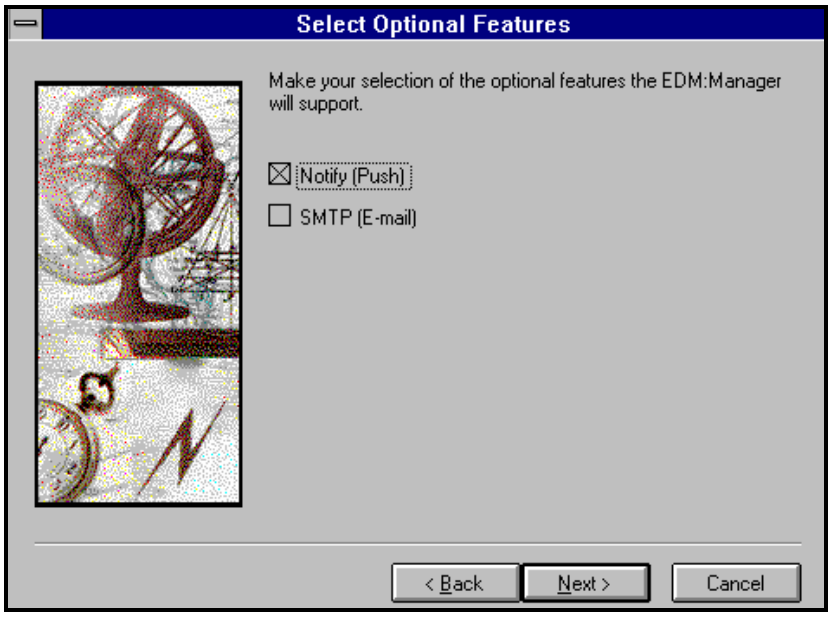
Specify a valid Manager name in the appropriate text box. This name can be up to 32 characters in length, and can include periods (.) and underscores (_).

The following dialog box is displayed:



10 Specify the appropriate communication protocol or combination of protocols and then choose **Next >**.

The following dialog box is displayed:.



11 Select one or more of the TCP/IP-supported features. Then, choose **Next >**.

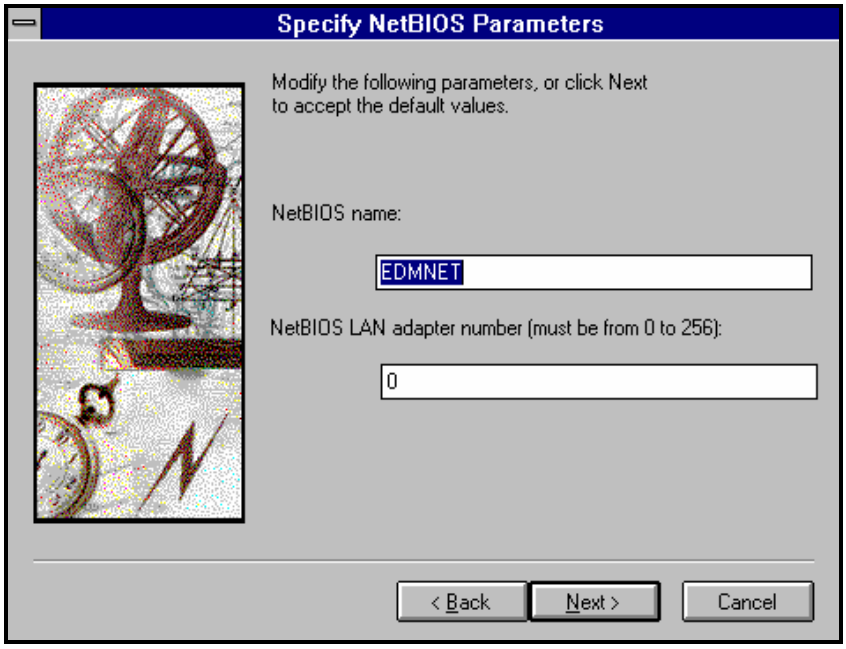
The **Notify (Push)** option will enable you to perform a specialized Manager-initiated EDM Client Connect Process.

The **SMTP (E-mail)** option will enable you to use the Simple Mail Transfer Protocol (SMTP) to send and receive EDM information.

Note: If you did not select the TCP/IP communication protocol, you can still select one or more of the TCP/IP-supported features if that protocol is on your desktop.

Depending on your communication protocols and optional features, one or more of the following screens is displayed.:

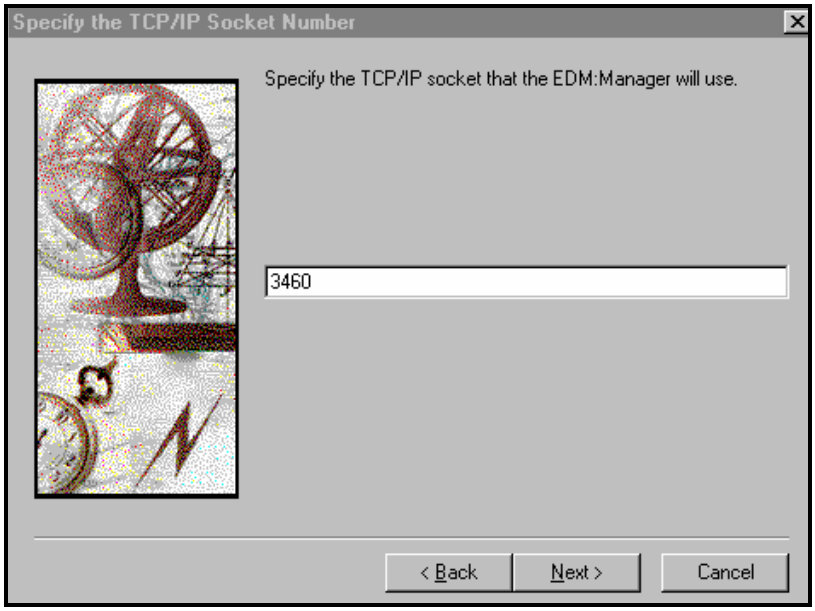
If you selected the **NetBIOS** communication protocol, the following dialog box is displayed:.



12 To accept the values shown, choose **Next >**. Or, specify valid values in the **NetBIOS name** and **NetBIOS LAN adapter number** text boxes, respectively.

Then, choose **Next >**.

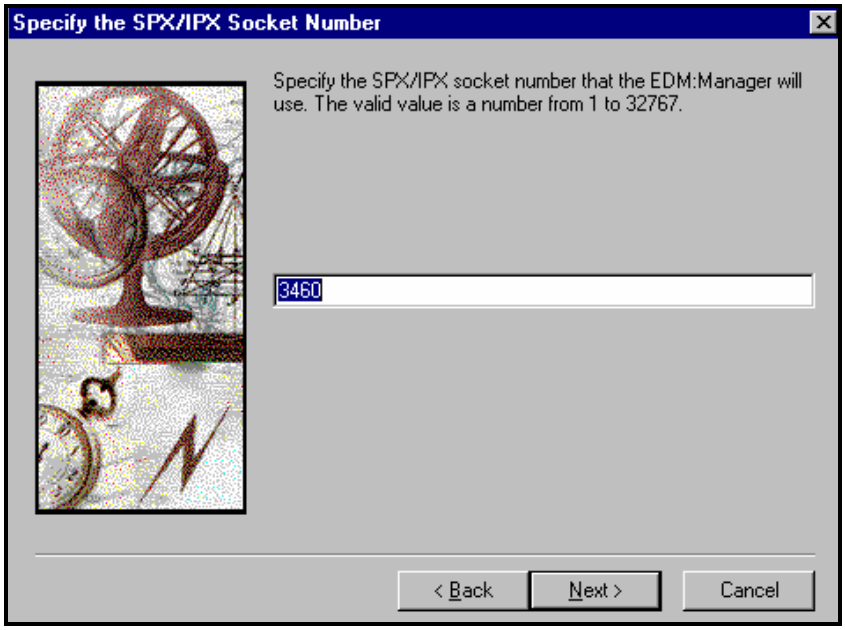
If you selected the **TCP/IP** communication protocol, the following dialog box is displayed:.



13 Specify the TCP/IP socket number the Manager is using. The value you specify here is based on your site's configuration. The default is **3460**.

Then, choose **Next >**.

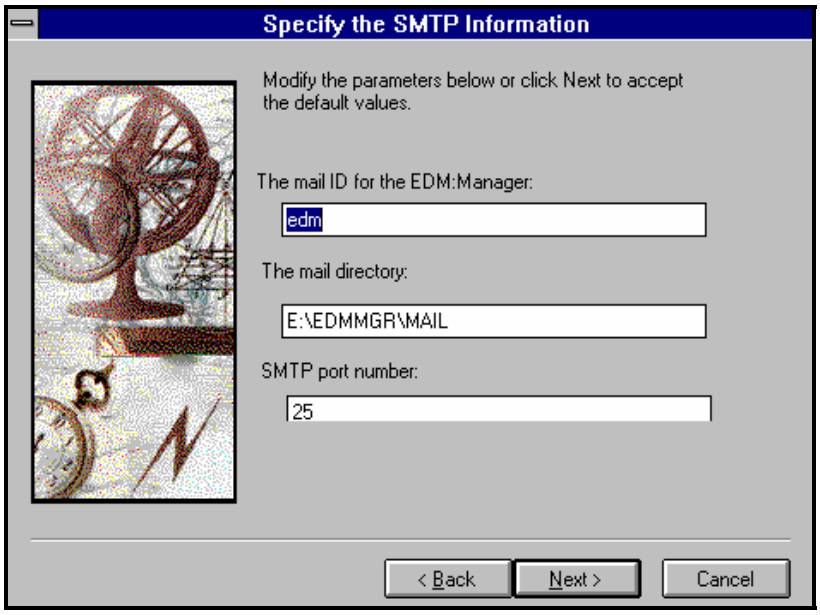
If you specified the **SPX/IPX** communication protocol, the following dialog box is displayed:.



14 Specify the appropriate SPX/IPX socket number. Then, choose **Next >**.

Depending on your communication protocols and optional features, one of the following dialof dialog boxes is displayed.

If you selected the **SMTP** optional feature, the following dialog box is displayed:.



15 You can accept the default values for the SMTP e-mail option, or specify other values by typing the information in the appropriate text boxes.

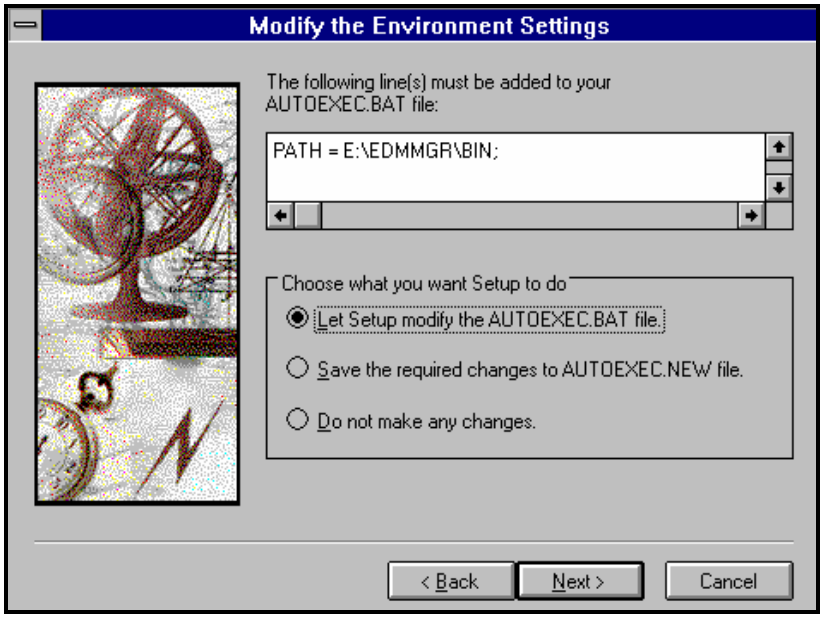
In **The mail ID for the EDM Manager:** text box, specify the appropriate mail ID so the Manager can accept the incoming mail addressed to this user ID, only.

In **The mail directory:** text box, specify the appropriate path for the mail directory. This directory queues outgoing mail.

In the **SMTP port number:** text box, specify the appropriate port number for your system. This port is where the SMTP receive Manager waits for the incoming mail. The default port number is **25**.

Then, choose **Next >** to continue the installation.

The following dialog box is displayed:.



16 If you want the install process to automatically update your AUTOEXEC.BAT file, accept the default value, and then choose **Next >**. Or, you can select one of the other available options.

If you select the option to save the required changes to the AUTOEXEC.NEW file, this file is placed in the same drive as the database that you installed earlier, for example:

```
F:\EDMDB\AUTOEXEC.NEW
```

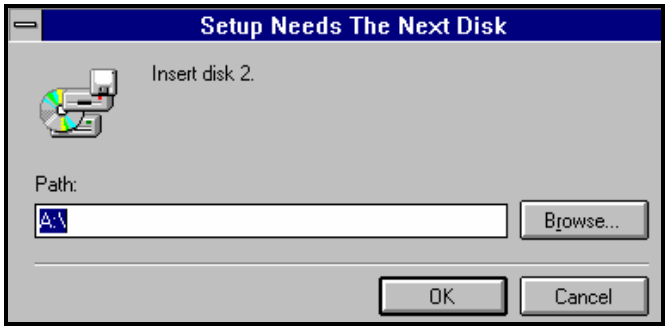
If you select the third option, **Do not make any changes**, a message is displayed stating that the AUTOEXEC.BAT file will not be modified. You must manually modify the file before rebooting your system and starting the EDM Manager.

If you want to make any changes to the previous screens before completing the installation, choose **< Back**.

Note: If you opted to install the Windows NT EDM Manager using an existing EDM Manager database, a message is displayed stating that the program is now cleaning up the previous EDM Manager database files.

If you opted to backup your existing EDM Manager database, a message is displayed stating that the program is now creating a copy of your existing database.

The following dialog box is displayed:.



17 Insert Disk 2, and then choose **OK**. The Setup screen specifies which installation files are being copied from each diskette.

Follow the prompts until you insert all of the requested installation diskettes.

Note: If you are installing the EDM Manager for Windows NT from CD-ROM, turn to step **17** now.

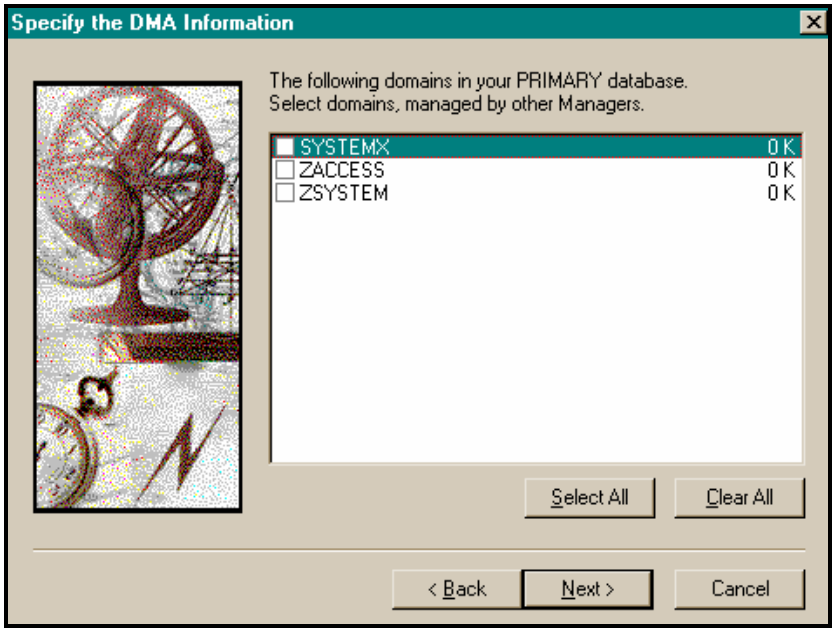
The following prompt is displayed:.

18 The program prompts you to specify whether you will be setting up the Distributed Manager Adapter (DMA) environment.

The program prompts you to specify whether you will be setting up the Distributed Manager Adapter (DMA) environment.

- If you choose **No**, you will not have to specify domain information. Turn to step **22.**
- If you choose **Yes**, then you will have to specify source EDM Manager information for the PRIMARY domains, as needed.

The following dialog box is displayed.



The list of domains displayed in this box include the domains in your PRIMARY database as well as the three EDM domains: **SYSTEMX**, **ZACCESS**, and **ZSYSTEM**.

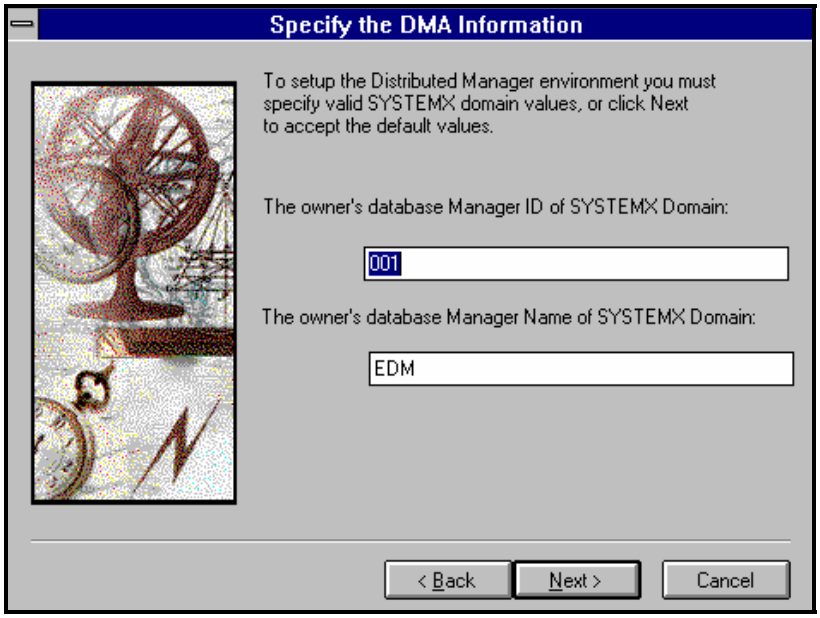
19 If you do not select any of the listed domains, the program will automatically specify your Manager ID as the source EDM Manager for all. Then, choose **Next. >** Turn to step **22** to continue.

- If you want to specify a different source EDM Manager for a domain, select the domain and then choose **Next >.** You will have to specify the source EDM Manager ID and Name for each domain that you select.
- If you do not select any of the listed domains, the program will automatically specify your Manager ID as the source EDM Manager for all. Then, choose **Next. >** Turn to step **22** to continue.
- If you want to specify a different source EDM Manager for a domain, select the domain and then choose **Next >.** You will have to specify the source EDM Manager ID and Name for each domain that you select.

Note: You can select all the domains by choosing **Select All**. If you choose this option, you will have to specify a source EDM Manager ID and Name for all of the domains listed.

If you want to deselect one or more of the domains, you can manually clear the domain check box or, choose **Clear All**.

Depending on your domain selections, one or more of the following dialog boxes is displayed.

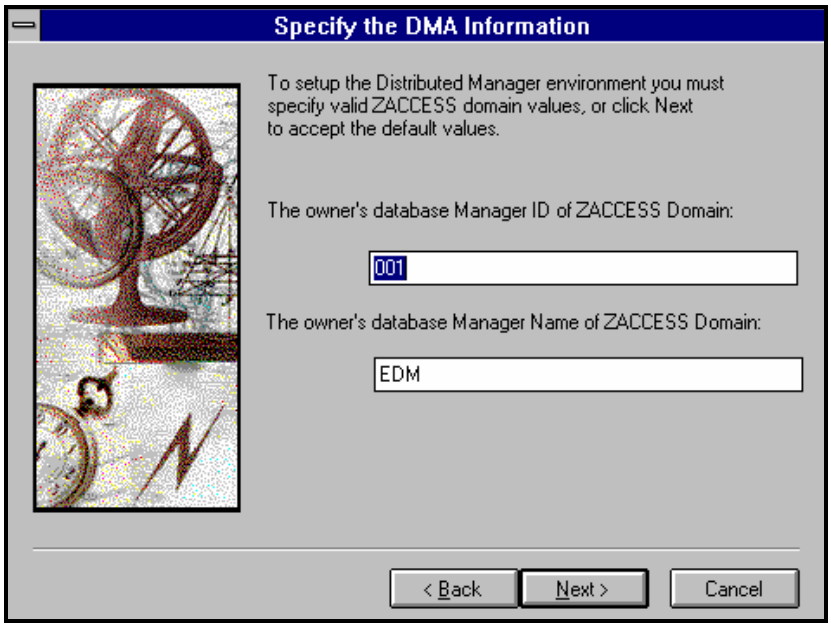


20 In this screen, you must specify a valid source Manager ID and Name for the SYSTEMX domain, to enable the DMA environment. Then, choose **Next >**.

In this screen, you must specify a valid source Manager ID and Name for the SYSTEMX domain, to enable the DMA environment. Then, choose **Next >**.

To specify different or additional domains, choose **< Back** to return to the domain listing.

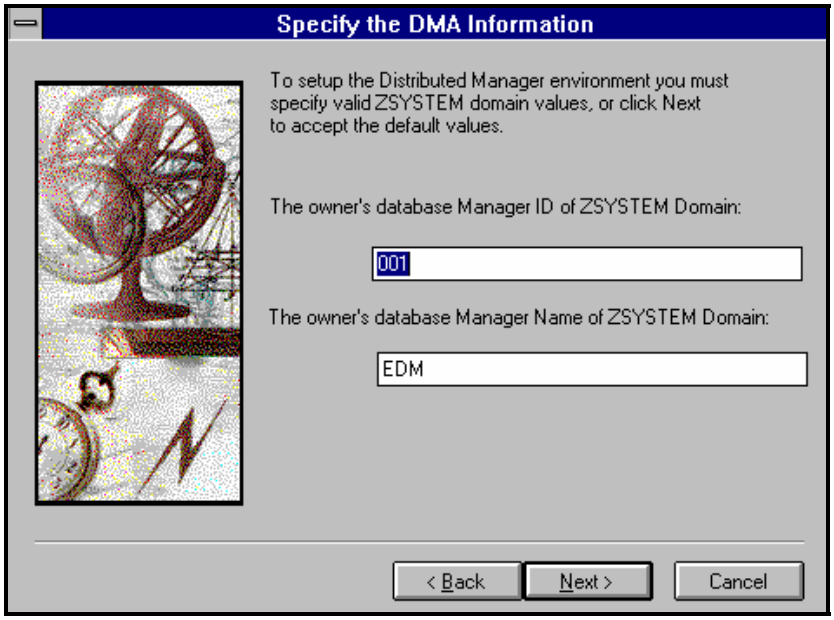
The following screen is displayed.:



21 In this screen, specify a valid source Manager ID and Name for the ZACCESS domain, to enable the DMA environment. Then, choose **Next >**.

In this screen, specify a valid source Manager ID and Name for the ZACCESS domain, to enable the DMA environment. Then, choose **Next >**.

The following dialog box is displayed:



22 In this screen, specify a valid database Manager ID and Name for the ZSYSTEM domain, to enable the DMA environment. Then, choose **Next >.**

Now that you have specified the parameters for the domains, you can select how you want to run the EDM Manager. The following prompt is displayed.

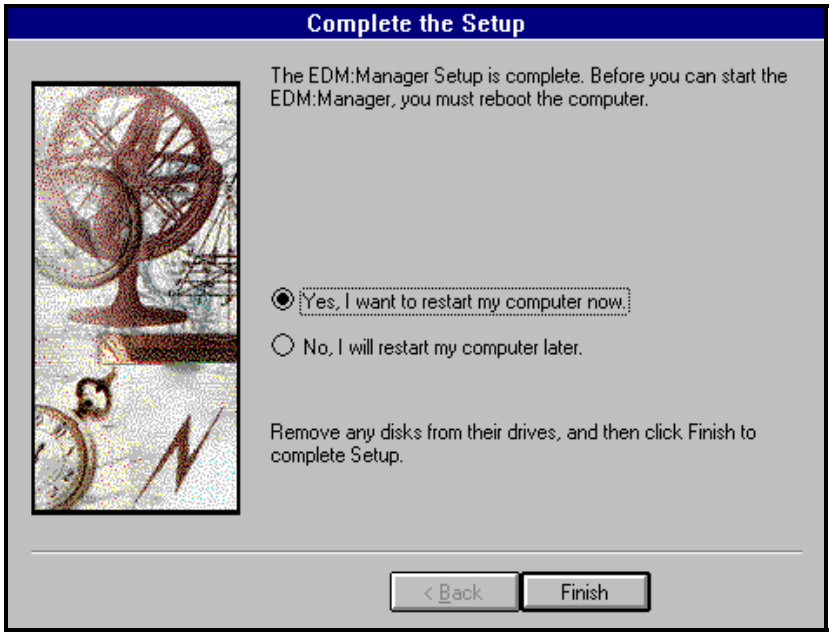
23 You are prompted to specify how you want to set up the EDM Manager; as an NT service or as a non-service.

Choose **Yes** to configure the EDM Manager as an NT service now.

Or, choose **No** to run the EDM Manager as a non-service, or as an NT service later.

Note: You can easily switch back and forth from NT service to non-service by shutting down the EDM Manager in the current service, and then starting it in the other service. For additional details, see see *"Chapter 2: Starting and Stopping the EDM Manager for Windows NT."* in Chapter 2.

The following dialog box is displayed:



24 Before running the EDM Manager, you must select one of the available options. Then, choose Finish to complete the installation.

If you want the EDM Manager to run as an NT service, select the option to restart your computer *now*.

The program will reboot your system and the EDM Manager will automatically be started.

If you want the EDM Manager to run as a non-service, select the option to restart your computer *later*. You will be returned to the Program Manager.

If you did not configure the EDM Manager to run as an NT service during the installation, you can run as a non-service by choosing the EDM Manager Startup icon. Or, you can configure the EDM Manager later to start as a manual NT service.

Warning: You must reboot your system before starting the EDM Manager.

For details, see *"Running the EDM Manager as a Non-Service"* and *"Running the EDM Manager as an NT Service"* later in this chapter.

For additional details, see "*Chapter 2: Starting and Stopping the EDM Manager for Windows NT*" in Chapter 2.

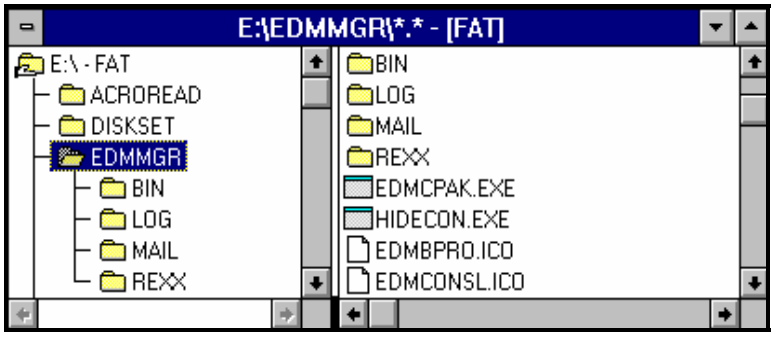
Note: If you want to change the service type after installing the EDM Manager for Windows NT, you must manually stop the Manager and then re-configure the service. For additional details, see *"Chapter 2: Starting and Stopping the EDM Manager for Windows NT."*

Note: During this installation, many directories were automatically created. Turn to page 38 39the next page to check the directory listing.

EDM Manager Directories

The EDM Manager directories are located in the File Manager in the directory you specified during step **5** of the EDM Manager for Windows NT installation for example, (E:\EDMMGR\).

Displayed below is a sample tree view of the directory and subdirectories that were created during the installation process. Depending on your specifications, the MAIL subdirectory will be created in the directory where you placed your EDM Manager files.



The table below lists and describes the directories that were automatically created during the EDM Manager for Windows NT installation process.

The EDM Manager Directory List

Directory Name	Description
BIN	This directory includes the EDM Manager file EDMPROF.DAT, database utilities, and all of the methods and executables for the EDM Manager, including the executables EDMNTSRV and ZTOPTASK. ZTOPTASK enables you to start the EDM Manager. This directory contains the DLL files for the EDM Manager.
LOG	The EDM Manager activity log is stored in this log directory.
MAIL	This directory contains the SMTP e-mail messages.
REXX	This directory contains the REXX methods.

Running the EDM Manager for Windows NT as a Non-Service

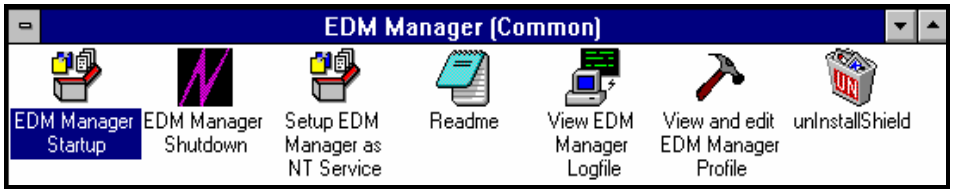
After you have completed the EDM Manager for Windows NT installation program, you can run the EDM Manager as a non-service.

> **Note**: If you have installed or are running the EDM Manager as an NT service, you must first stop the EDM Manager NT service, and then continue with the following steps. For additional details, see "*Chapter 2: "Chapter 2: Starting and Stopping the EDM Manager for Windows NT."* in Chapter 2.

➢ **To Run the EDM Manager for Windows NT as a Non-Service (Version 3.51):**

1 Choose the **EDM Manager** icon located in the **Program Manager**.

The **EDM Manager (Common)** group box is displayed:

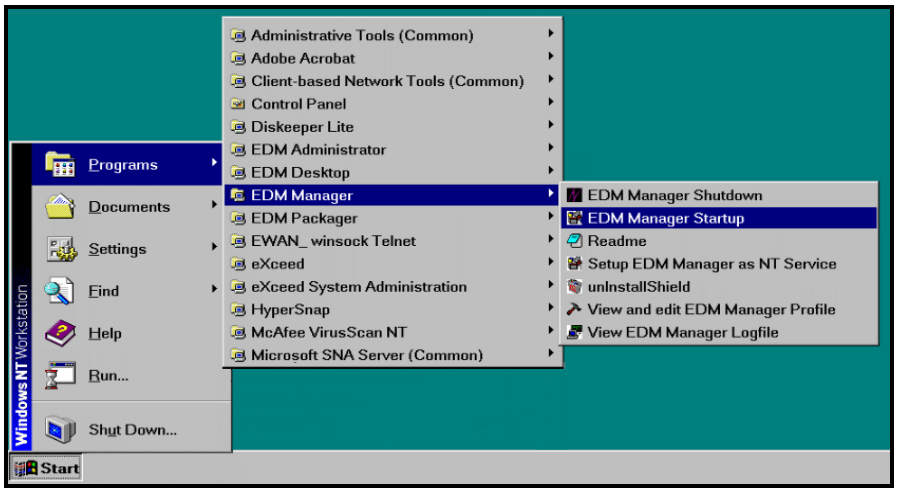


2 Choose the **EDM Manager Startup** icon.

➢ **To Run the EDM Manager for Windows NT as a Non-Service (Version 4.0):**

1 From the **Start** menu, select **Programs.**

2 From the Programs submenu, select EDM Manager.



3 Choose EDM Manager Startup.

The EDM Manager is started. You can verify that the EDM Manager has started by viewing pertinent messages in the EDM Manager log.

The command prompt is displayed on the screen, indicating that the EDM Manager is running as a non-service.

For additional details, see *"Chapter 2: Chapter 2: Starting and Stopping the EDM Manager for Windows NT"* in Chapter 2.

Running the EDM Manager for Windows NT as an NT Service

To run the EDM Manager as an NT service, you must configure it as an NT service.

If you configured the EDM Manager as an NT service during installation and rebooted the system, then the EDM Manager service is probably already started. (To check this, choose the **Services** icon from the **Control Panel**.)

If you run the EDM Manager as an NT service, you have the option to configure the EDM Manager startup process. This service lets you run the EDM Manager manually or automatically upon rebooting your system, as well as control the starting and stopping of a service.

Once the service is running, you can pause the service at any time during that session, and then continue at the point where you left off.

For additional details on configuring the NT service, starting, and stopping see "*Chapter 2: Starting and Stopping the EDM Manager for Windows NTT.*" in Chapter 2.

2 Starting and Stopping the EDM Manager for Windows NT

This chapter explains how to configure the EDM Enterprise Manager to automatically start during Windows NT startup processing, and how to manually start and stop the EDM Manager for Windows NT if it is running as either an NT-service or a non-service. This chapter also describes how to use the Windows NT Event Log to view key EDM Manager messages.

Configuring EDM Manager for Windows NT Startup

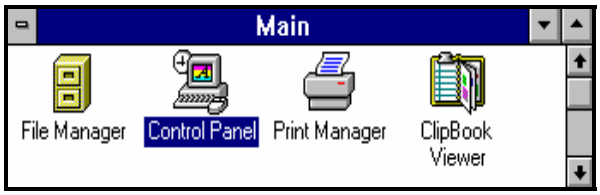
When the Manager is run as an NT service, you can configure it to initialize automatically during startup processing, or to be started manually. In addition, you can disable the Start and Stop buttons on the Services panel.

To run the Manager as a non-service, see "*Starting Starting and Stopping the EDM Manager in for Windows NT as a Non-Service*" in this chapteron page 50.

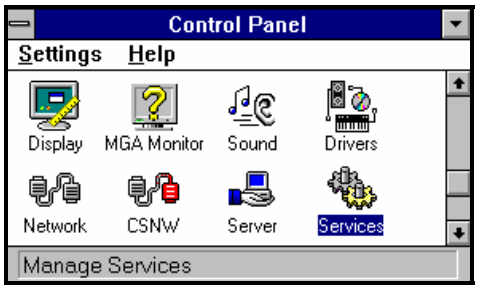
Windows NT 3.51

➢ **To Configure the EDM Manager as an NT Service:**

1 Choose the **Main** icon from the Windows NT program group. The Main program group is shown below.



2 Choose the **Control Panel** icon. The Control Panel program group is shown below.



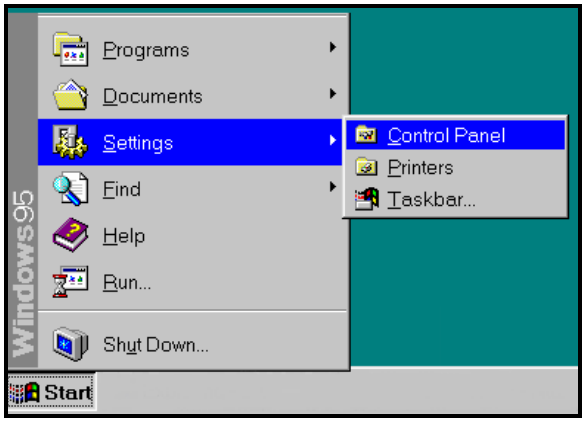
3 Choose the **Services** icon.

4 Go to step **4** in the section "*Windows NT 4.0"* below.

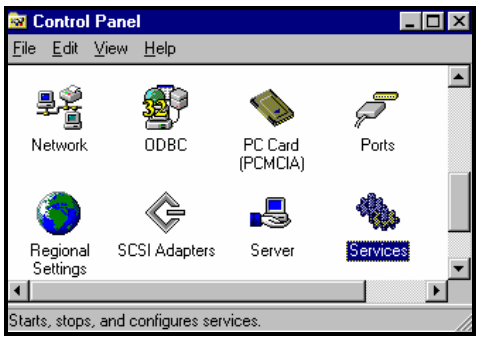
Windows NT 4.0

➢ **To Configure the EDM Manager as an NT Service:**

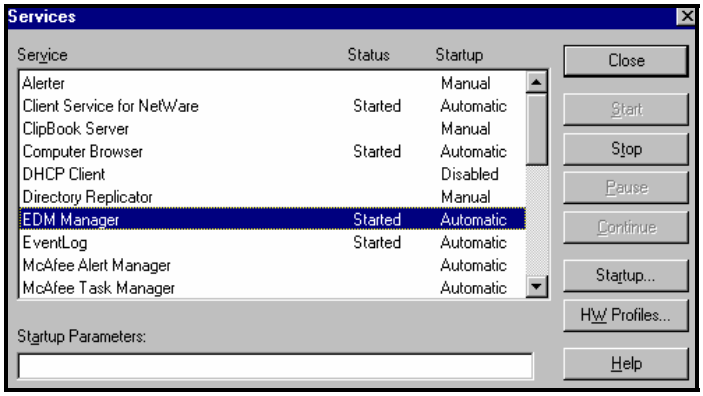
1 From the **Start** menu, select **Settings**.



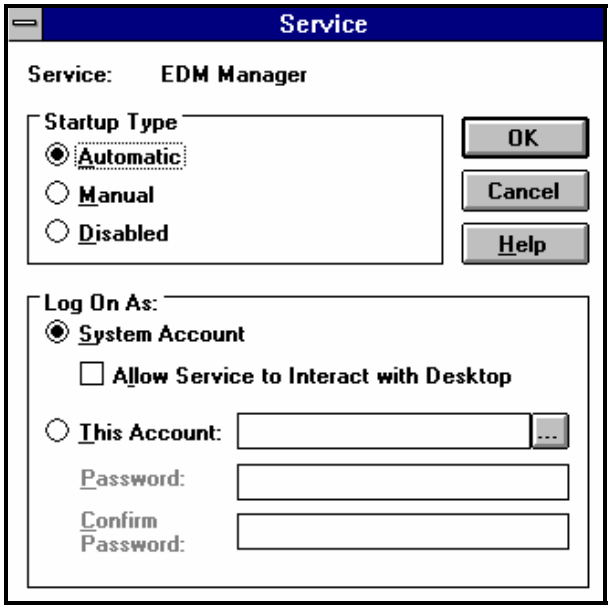
2 From the **Settings** submenu select **Control Panel**. The Control Panel program group is shown below.



4 From the Control Panel program group choose the **Services** icon. The **Services** dialog box is shown below.



5 Choose the **EDM Manager** entry from the **Services** dialog box. The **Service** dialog box is displayed:



6 The **Startup Type** dialog box has three configuration options: **Automatic, Manual**, and **Disabled.**

The table below lists and describes these configuration options.

Configuration Options

If you want to	Select
Automatically start the EDM Manager for Windows NT when the system is Initially booted or subsequently rebooted. Note if the EDM Manager is configured as Automatic, no startup actions are necessary to run the EDM Manager as an NT service.	**Automatic**
Manually start the EDM Manager. For details, see *"Starting and Stopping the "To start the EDM Manager foras a manual Windows NT service as a Service""* on page 44..	**Manual**
Disable the Start and Stop buttons on the Services panel. You can neither start nor stop the EDM Manager from the Services panel while it is disabled. To re-enable these buttons, return to the Services panel and select Startup. The Service dialog box is displayed. You may select either the Automatic or Manual configuration option. In either case, the Start and Stop buttons are now enabled.	**Disabled**

Note: You can change the type of startup (for example, Automatic, Manual, or Disabled) for the EDM Manager entry by returning to the Services panel and changing the configuration option.

After making a change to the startup option, you must reboot the system for the change to take effect.

Starting and Stopping the EDM Manager for Windows NT as a Service

The EDM Manager for Windows NT can be run as an automatic NT service, a manual NT service, or a non-service. The method that you choose determines how you start, stop, and monitor the EDM Manager for Windows NT.

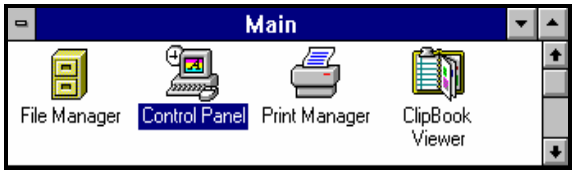
If you have configured your EDM Manager for Windows NT as a service and selected Automatic as your Configuration Option, the Manager will be automatically initialized during Windows NT startup processing. You can verify that the Manager has started by viewing pertinent messages in both the NT Event Log and the Manager log.

If you have configured your EDM Manager for Windows NT as a service and selected Manual as your Configuration Option, the EDM Manager will not be automatically initialized during Windows NT startup processing.

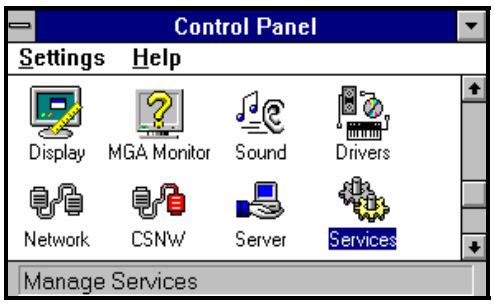
Starting the EDM Manager as a Manual Service in Windows NT 3.51

➢ **To Start the EDM Manager:**

1 Choose the **Main** icon from the Windows NT program group. The Main program group is shown below.



2 Choose the **Control Panel** icon. The Control Panel program group is shown below.



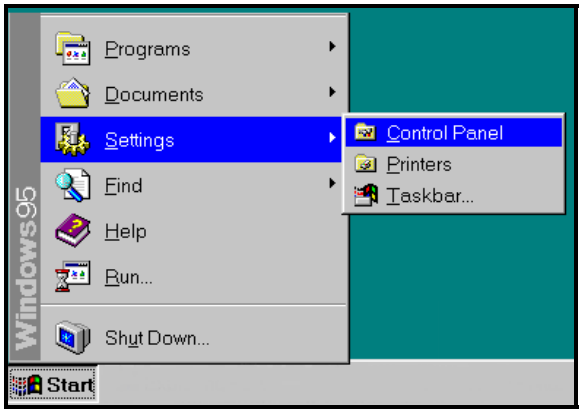
3 Choose the **Services** icon.

4 Go to step **4** in the section *"Starting the EDM Manager as a Manual Service in Windows NT 4.0"* below.

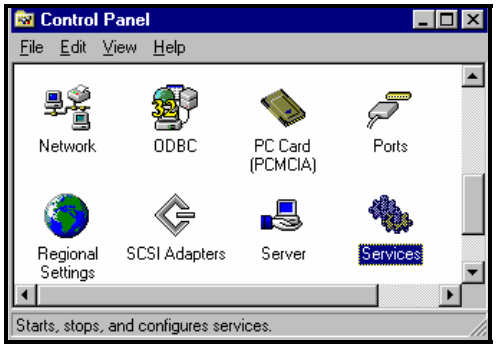
Starting the EDM Manager as a Manual Service in Windows NT 4.0

➢ **To Start the EDM Manager:**

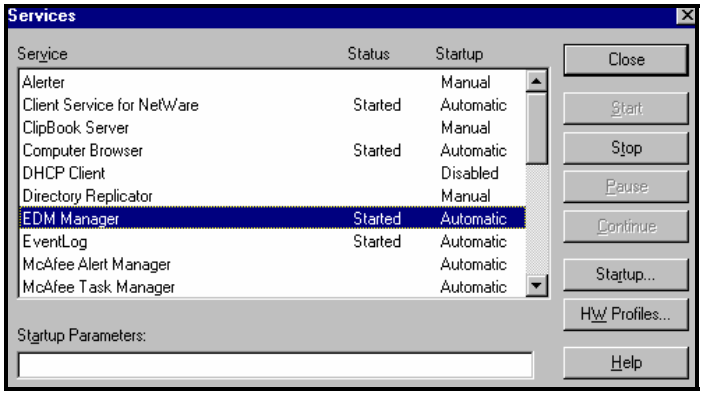
1 From the **Start** menu, select **Settings**.



2 From the **Settings** submenu select **Control Panel**. The Control Panel dialog box is shown below.



3 Choose the **Services** icon. The Services dialog box is shown below.



4 Choose the **EDM Manager** entry from the Services dialog box.

5 Choose **Start.**

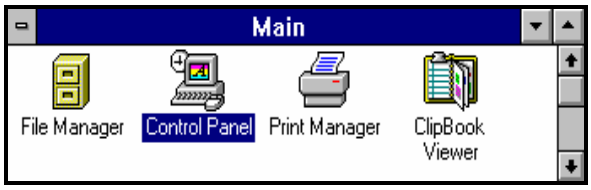
The status field in the Manager line changes from blank to **Started**. You can verify that the Manager has started by viewing pertinent messages in both the NT Event Log and the Manager log.

If you have not configured your EDM Manager for Windows NT as a service, or choose to run the Manager as a non-service, you must manually start the Manager.

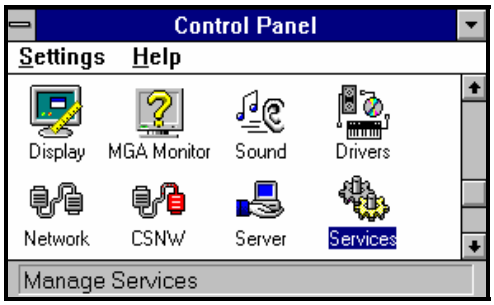
Stopping the EDM Manager in Windows NT 3.51 When Running as a Service

➢ **To Stop the EDM Manager:**

1 Choose the **Main** icon from the NT program group. The Main program group is shown below.



2 Choose the **Control Panel** icon. The Control Panel program group is shown below.



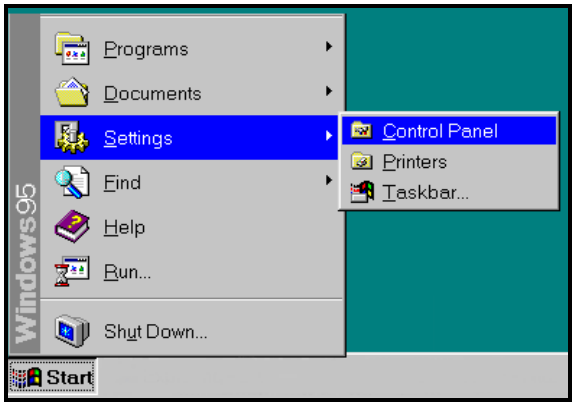
3 Choose the **Services** icon.

4 Go to step **4** in the section *"Stopping the Manager in Windows NT 4.0 while When it is Running as a Service"* on the next page.*e."*

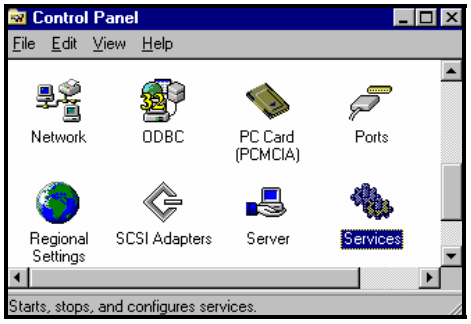
Stopping the EDM Manager in Windows NT 4.0 When Running as a Service

➢ **To Stop the EDM Manager:**

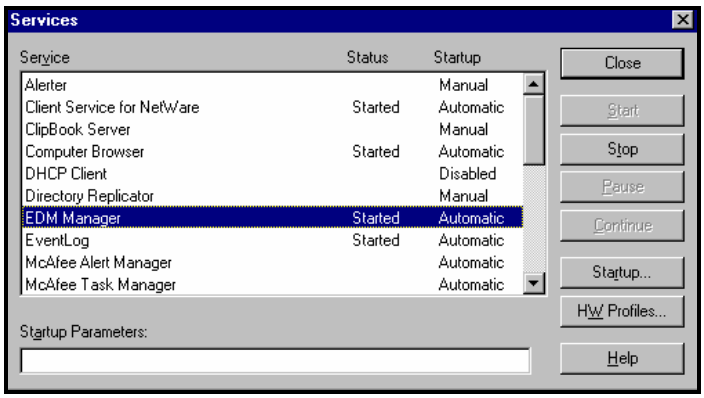
1 From the **Start** menu, select **Settings.**



2 From the **Settings** submenu select **Control Panel**. The Control Panel dialog box is shown below.



3 Choose the **Services** icon from the **Control Panel** dialog box. The Services dialog box is shown below.



4 Choose the **EDM Manager** entry from the **Services** dialog box.

5 Choose **Stop.**

The program will verify that you want to stop the Manager.

When you choose **Yes** at that prompt, the Manager is stopped, and the Manager status field changes from **Started** to **Stopped**.

Starting and Stopping the EDM Manager for Windows NT as a Non-Service

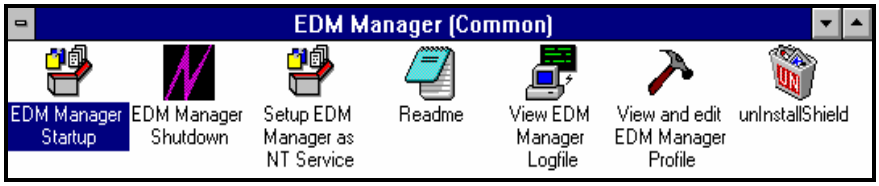
The EDM Manager for Windows NT can be run as a non-service. The method you use to start, stop, and monitor the EDM Manager for Windows NT will be outlined in this section.

If you have configured your EDM Manager for Windows NT as a non-service, you must manually start the EDM Manager after Windows NT startup processing.

Starting the EDM Manager in Windows NT 3.51

➢ **To Start the EDM Manager as a Non-Service:**

1 Choose the **EDM Manager** icon from the Windows NT program group. The EDM Manager program group is shown below.



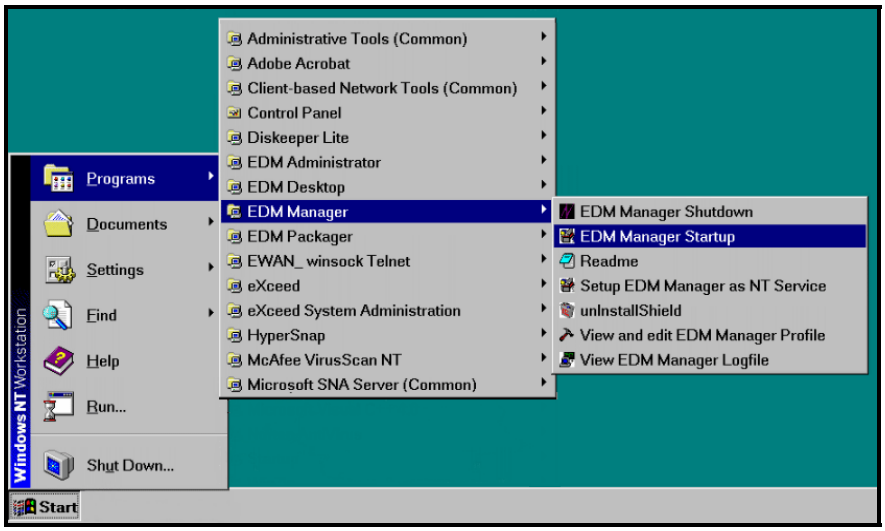
2 Choose the **EDM Manager Startup** icon to run the EDM Manager program.

Starting the EDM Manager in Windows NT 4.0

➢ **To Start the EDM Manager as a Non-Service:**

1 From the **Start** menu, select **Programs**.

2 From the **Programs** submenu, select **EDM Manager**.



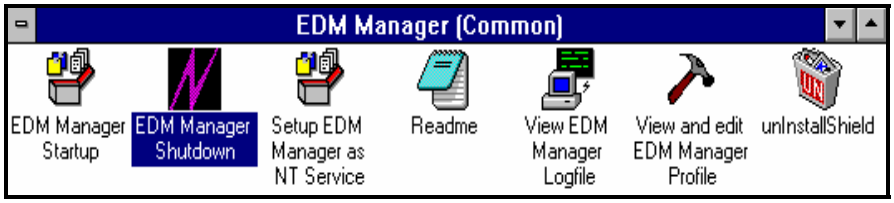
3 Choose **EDM Manager Startup**.

The Manager is started. You can verify that the Manager has started by viewing pertinent messages in the Manager log.

Stopping the EDM Manager in Windows NT 3.51

➢ **To Stop the EDM Manager:**

1 Choose the **EDM Manager** icon from the NT program group. The EDM Manager program group is shown below.



2 Choose the **EDM Manager Shutdown** icon.

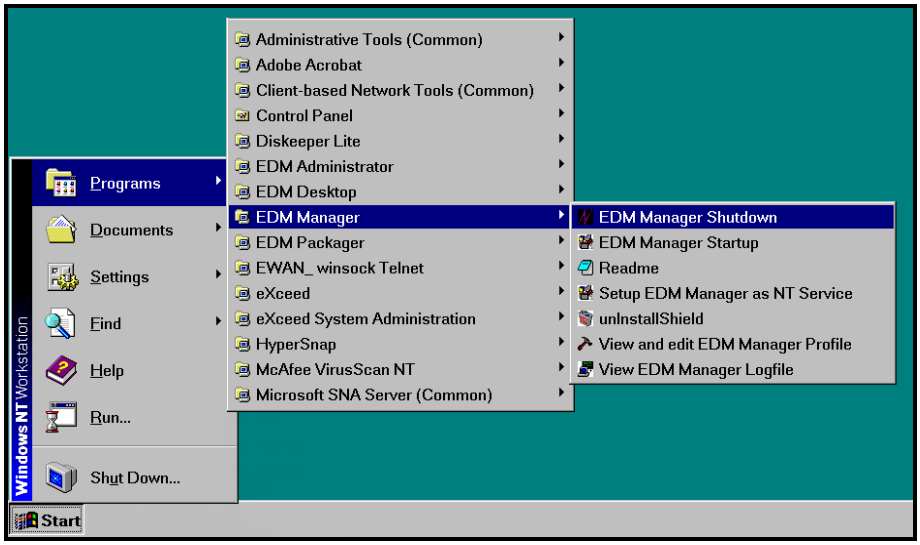
The Manager has shut down. You can verify that the Manager has shutdown by viewing pertinent messages in the Manager log.

Stopping the EDM Manager in Windows NT 4.0

➢ **To Stop the EDM Manager as a Non-Service:**

1 From the **Start** menu, select **Programs.**

2 From the **Programs** submenu, select **EDM Manager**.



3 Choose **EDM Manager Shutdown**.

The Manager is stopped, and a series of shutdown messages is displayed.

Using the Windows NT Event Log with the EDM Manager

Windows NT allows you to monitor the Manager startup process, errors in the startup process, and Manager shutdown through the NT Event Log. The Event Log is enabled only when running the Manager as an NT service. You can view the Event Log by opening the Event Viewer.

Accessing the Event Viewer in Windows NT 3.51

➢ **To Access the Event Viewer:**

1 Choose the **Administrative Tools** icon from the NT program group. The Administrative Tools program group is shown below.

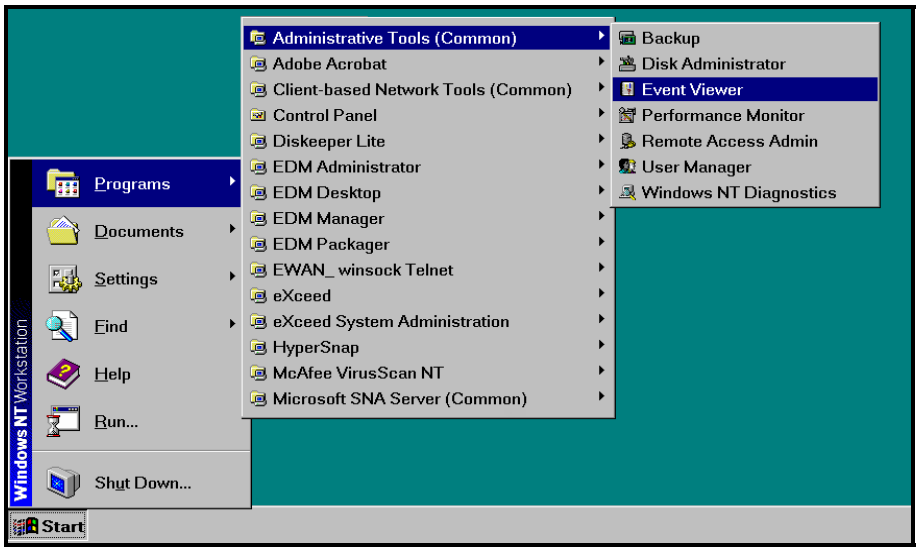
2 Choose the **Event Viewer** icon.

Accessing the Event Viewer in Windows NT 4.0

➢ **To Access the Event Viewer**

1 From the **Start** menu, select **Programs.**

2 From the **Programs** submenu, select **Administrative Tools (Common).**



3 Choose **Event Viewer.**

The Event Log is displayed.

Event Viewer Messages

The Event Log produces five EDM-related messages.

To determine the meaning of the message, highlight the entry line (which is identified in all cases by ZTOPTASK), and select it by using the mouse.

The five messages produced by the Event Log, and their meanings, are:

Startup Message

```
SERVICE HAS STARTED
```

The EDM Manager for Windows NT is beginning its startup processing by running the ZTOPTASK program.

Error Message 1

```
START SERVICE CONTROL DISPATCHER FAILED
```

The EDM Manager for Windows NT failed during service initialization. Startup processing has stopped.

Error Message 2

```
SERVICE CONTROL HANDLER REGISTRATION FAILED IN SERVICE MAIN
```

During second stage initialization, the NT service controller failed to update the Registry Keys. Startup processing has stopped.

Error Message 3

```
SET SERVICE FAILED IN REPORT STATUS
```

An attempt was made to report the status of the EDM Manager for Windows NT startup process to the Service Manager. However, the Service Manager did not receive the status report.

The Manager service may actually be running, or the startup service has stopped. The Service Manager is unable to discern what the actual condition is.

Shutdown Message

```
SERVICE STOPPED BY STOP REQUEST OR SHUT-DOWN REQUEST
```

The Manager has shut down.

3 Tuning the EDM Manager for Windows NT

This chapter shows you how to tune the EDM Manager for Windows NT by working with the sections of the EDMPROF.DAT file.

Understanding the Tuning Process

The sections of the EDMPROF.DAT contain the operational parameters of the EDM Manager. You can tune many aspects of the EDM Manager by working with the settings in the various EDMPROF.DAT sections.

The performance of the EDM Manager depends on a number of factors, including the number of concurrent EDM Clients being processed, the complexity of the configurations for those EDM Clients, the volume of the data being processed, and so forth. The configuration of the EDM Manager log, which documents system status for informational purposes and for problem determination, can also alter performance.

This chapter describes your options for working with each section of the EDMPROF.DAT file.

> **Note**: There are many factors outside of the EDM Manager that affect the EDM Manager's performance, such as how many other applications are running on the same machine and processes running in the same processor, the speed of the processor, the bandwidth of the network, the number of processors, and so forth.
>
> You cannot tune these factors by adjusting the EDMPROF.DAT settings. Instead, you must tune these factors outside of the EDM Manager.

Viewing and Editing the EDM Manager Profile

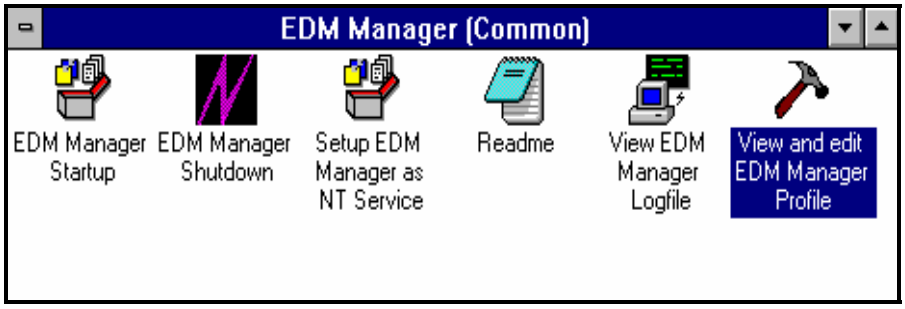
You can adjust the parameters that comprise the EDM Manager's EDMPROF.DAT file for Windows NT. The EDMPROF.DAT file is located in the BIN subdirectory of the EDMMGR directory.

The following instructions explain how to view and edit the EDM Manager profile.

➢ **To View and Edit the EDM Manager Profile (Windows NT 3.51)**

1 Choose **EDM Manager** located in the Program Manager.

The **EDM Manager (Common)** window is displayed:



2 Choose the **View and edit EDM Manager Profile** icon.

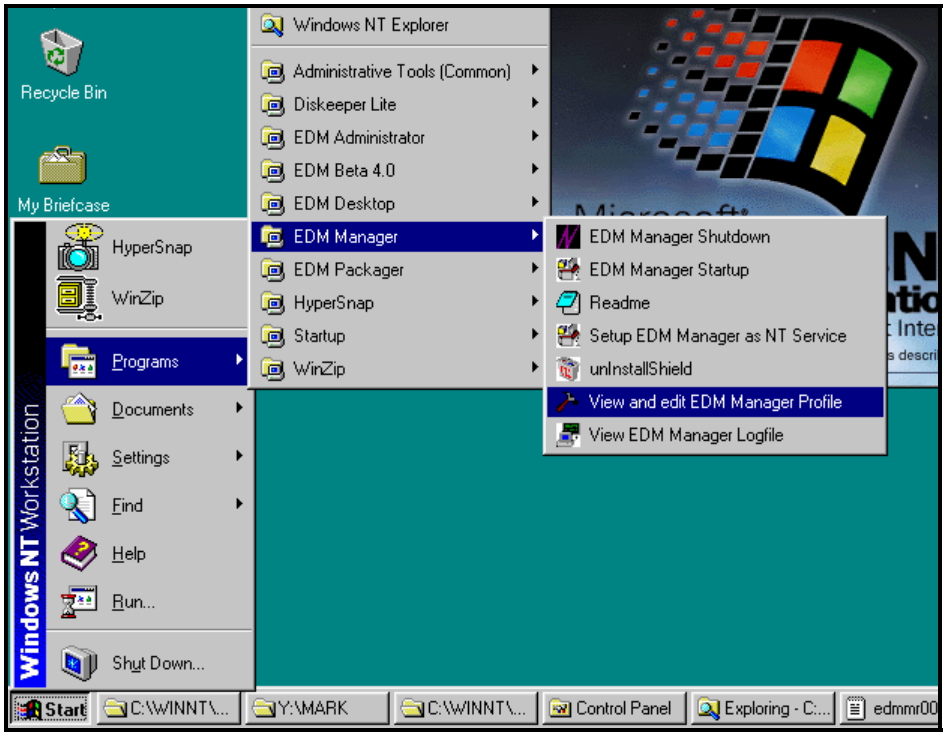
The EDMPROF.DAT panel is displayed.

➢ **To View and Edit the EDM Manager Profile (Windows NT 4.0)**

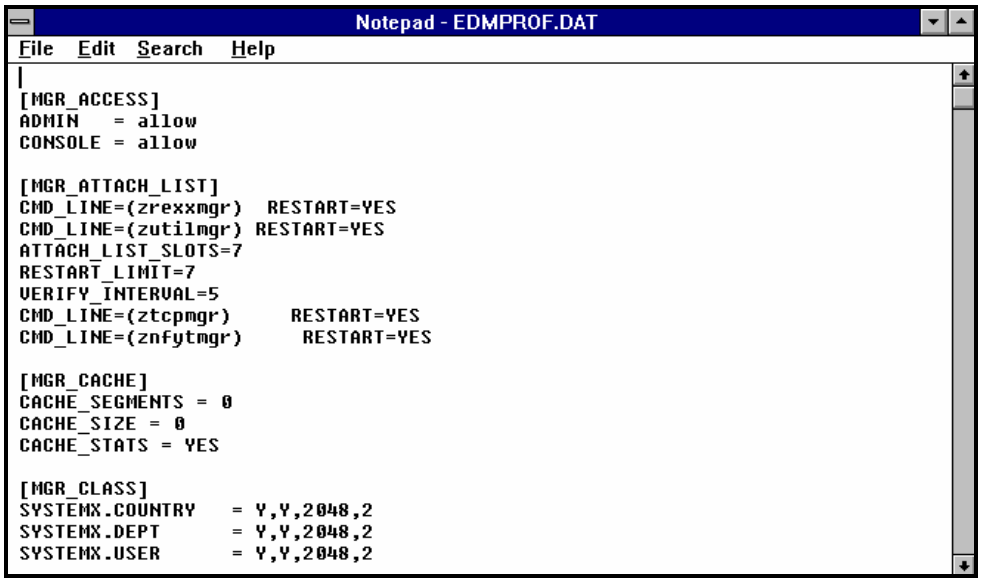
1 Choose the **Programs** folder from your **Start** menu.

2 Then, select **EDM Manager** from the next program list.

3 Choose **View and Edit the EDM Manager Profile**.



The EDMPROF.DAT panel is displayed.



This scrollable panel displays the EDM Manager profile of operational settings in the EDM Manager.

To change a specific setting, highlight the appropriate information, and type in the new value on top of the old one, or use the DELETEELETE key to erase the setting.

Warning: Once you change a setting within this file, you must restart the EDM Manager to invoke the changed settings.

To close the EDM Manager Profile, choose **Exit** from the **File** menu.

If you made changes to the EDMPROF.DAT file, a message will prompt you to save your changes. Choose **Yes**.

The EDMPROF.DAT File at a Glance

The EDMPROF.DAT file contains the parameters that determine how your EDM Manager for Windows NT is configured to operate. The various sections within the EDMPROF.DAT file establish the settings that control the following different aspects of the EDM Manager:

- **Identification**

  The MGR_STARTUP section specifies the Manager by ID, name, type, and communications port.

  MGR_DIRECTORIES identifies the directory paths for the EDM data bases, EDM REXX Methods, and EDM non-REXX methods.

  The MGR_LICENSE contains your unique NOVADIGM license number.

- **Specification**

  The EDMPROF.DAT file establishes application-wide technical parameters. MGR_CACHE contains cache processing options, while MGR_TPINIT identifies communications packet sizes.

- **Initialization**

  The MGR_ATTACH section allows you to define which Manager programs are initiated at startup, as well as options for their functioning. The MGR_CLASS section specifies which classes and instances of the EDM data base are cached during the Manager initialization process.

- **Operation**

  A number of EDMPROF.DAT sections contain parameters for EDM system operations. MGR_ACCESS determines Manager access to EDM Administrator and Manager Console functions. MGR_METHODS identifies options for method processing. The MGR_NOTIFY section specifies the defaults for the Manager notification of EDM Clients. MGR_OBJECT_RESOLUTION identifies a method for handling the ZADMIN object, for compatibility with pre-Version 3.0 data bases. MGR_RETRY values determine when an EDM Client will attempt to reconnect with the Manager. MGR_TASK_LIMIT identifies the parameters for concurrent Manager tasks, both ongoing and deferred. The MGR_TIMEOUT section specifies the amount of time a Manager will wait for an inactive Client. MGR_SNMP runs the platform that processes SNMP requests and SNMP requests in turn are used to query or modify certain variables.

- **Monitoring**

  The MGR_LOG and MGR_TRACE sections identify where the EDM Manager log is located, how 'elastic' it is, and which individual system traces are being captured in the log. You can specify the parameters for using SMTP mail messages to support Manager monitoring in the MGR_SMTP_MAIL section.

EDMPROF.DAT File Settings

Most of the settings within the EDMPROF.DAT file's sections may be uniquely configured, and some are based on operating system and communications requirements. Also, while many of the sections are optional, a number are required for proper Manager functioning.

> **Note**: Since the EDMPROF.DAT file contains optional sections and may be uniquely configured to your particular requirements, you may not see every setting represented when you view your EDMPROF.DAT file.

The EDMPROF.DAT file is created during the installation of the EDM Manager. In many cases, you are prompted to enter information which is put directly into the EDMPROF.DAT file. Some values are derived from installation options, while other parameters are automatically entered during the installation.

There are two types of values in the EDMPROF.DAT file. The "As Installed" value represents a manual input or a derived entry. The "Default value" is the entry established by the EDM Manager if there is a blank value for that setting.

The table below provides a list of each section, and a brief description. This table also tells you whether the section is a required or optional section.

The EDM Manager for Windows NT Section List

Section Name	Description	Required or Optional
MGR_ACCESS	Specifies access to the EDM Administrator and the EDM Console function.	Optional
MGR_CACHE	Specifies cache processing options, such as cache segments, size, and statistics.	Optional
MGR_CLASS	Specifies the classes and class instances to be cached at Manager startup.	Optional
MGR_DIRECTORIES	Specifies the path for the EDM data bases, REXX methods, and non-REXX methods.	Recommended but not required
MGR_LICENSE	Enforces the licensing of EDM.	Required
MGR_LOG	Specifies the logging directory and logging options for the EDM Manager logging facility.	Recommended but not required
MGR_METHODS	Specifies options for method execution.	Recommended but not required
MGR_NOTIFY	Specifies the options for notify processing of TCP/IP EDM Clients.	Optional
MGR_OBJECT_ RESOLUTION	Specifies a method to handle the ZADMIN object.	Optional
MGR_RETRY	Specifies when an EDM Client should attempt a re-connection with the EDM Manager following rejection.	Optional
MGR_SMTP	Specifies the parameters the EDM Manager uses to interface with SMTP.	Optional
MGR_SNMPMGR_OBJECT_ RESOLUTION	Specifies the program where a platform is run processing SNMP requestsSpecifies a method to handle the ZADMIN object.	Recommended but not requiredOptional
MGR_STARTUP	Specifies the EDM Manager ID and TCP/IP port number.	Required
MGR_TASK_LIMIT	Specifies the number of concurrent tasks allowed.	Required
MGR_TIMEOUT	Specifies how long the EDM Manager will wait for a request from a connected EDM Client before disconnecting it.	Optional
MGR_TPINIT	Specifies packet sizes to send partners, and the maximum record size.	Required, should not be changed except by EDM support.
MGR_TRACE	Controls and influences diagnostic logging for the EDM Manager.	Optional
MGR_USERLOG	Specifies the logging directory and options for the EDM User logging facility	Optional

The following sections provide a description of each EDMPROF.DAT section, as well as the individual settings and the possible values for each setting. Also, the impact of tunable settings within each section on EDM Manager performance is described.

MGR_ACCESS

The MGR_ACCESS section of the EDMPROF.DAT file specifies access for the EDM Administrator and EDM Console.

The MGR_ACCESS section has the following two settings:

Setting Name	Description
ADMIN	Specifies access for the EDM Administrator. The value is DENY, ALLOW or IGNORE for this option. *Value as Installed:* DENY *Default Value:* DENY
CONSOLE	Specifies access for the EDM Console. The values are DENY or ALLOW for this option. *Value as Installed:* DENY *Default Value:* DENY

Example

```
*-----------------------------------------------------------*

* MGR_ACCESS SECTION                                       *

* THIS SECTION ALLOWS YOU TO SPECIFY THE DEFAULT ACCESS     *

* VALUES FOR THE EDM ADMINISTRATOR AND EDM CONSOLE.  THE    *

* DEFAULTS FOR CONSOLE AND ADMIN ARE DENY.                  *

*-----------------------------------------------------------*

[MGR_ACCESS]

ADMIN   = DENY

CONSOLE = DENY
```

Setting Name	Value as Installed	Default Value
ADMIN	DENY	DENY
CONSOLE	DENY	DENY

Performance Considerations

The following are the performance considerations associated with the MGR_ACCESS section:

- A value of ALLOW in the MGR_ACCESS setting will provide access to the EDM Administrator without any check of the ZACCESS domain of the data base. The result of setting ADMIN=ALLOW is as follows If the Administrator attempts an action (for which no access rules have been defined) the attempted action will be allowed.:.

- Access rules governing administrator actions are defined within the ZACCESS domain of the data base when the value specified is DENY.The result of setting ADMIN=DENY is as follows: The Administrator will be unable to perform an action unless there is an access rule specifically allowing such an action.

- The result of setting ADMIN=IGNORE is as follows: The Administrator will be able to perform any action because the access rules will be ignored by the EDM:Manager. Setting the Option to IGNORE essentially disables all Access rules as they relate to Adminitrator functions.

- Access to EDM Console is determined by local password security policy and also may be controlled by NT security features.

MGR_ATTACH_LIST

The MGR_ATTACH_LIST section of the EDMPROF.DAT file specifies which programs (Manager tasks) are to be attached when the EDM Manager is started. This section enables you to specify options for these processes.

The MGR_ATTACH_LIST section has the following five settings:

Setting Name	Description
ATTACH_LIST_SLOTS	Number of slots for attach list kept in shared memory of EDM Manager. Every entry is 132 bytes long. We recommend that you specify this parameter as the number of cmd_lines in this section plus 1.
RESTART_LIMIT	Number of attempts to restart an attached process that has terminated.
VERIFY_INTERVAL	Interval (in minutes) for the Task Manager to verify that attached processes are still running. In minutes. If this value is 0, no verification will occur.
CMD_LINE =	Command line to use in order to start the requested task. No blanks in substring 'CMD_LINE=( ) are allowed.
RESTART =	Switch to determine if a manager task will be restarted by the EDM Manager when abnormally terminated. Values for this setting are YES or NO. No Spaces should appear in the "RESTART=YES/NO" string.

Example

```
*ATTACH_LIST SECTION                                          *

*-----------------------------------------------------------*

*ATTACH_LIST = List of programs to attach when the EDM        *

 *Manager starts. The ztcpmgr, zrexxmgr, and znfytmgr         *

*processes are recommended. No process starts are required,   *

 *however, system ability is limited if the processes         *

 * *mentioned above are attached.                             *

*CMD_LINE    = Command line to use when starting process -    *

*no blanks in substring 'CMD_LINE=(' are allowed.             *

*RESTART=YES = If abnormally terminated will be restarted by *

*EDM Manager.  The default is NO.is if RESTART=YES is not     *

*specified. No specs are allowed in the "RESTART=YES/NO"      *

 string                                                      **

*ATTACH_LIST_SLOTS= Number of slots for attach list kept in   *

*shared memory of EDM Manager. Every entry is 132 bytes       *

*long, we recommend that you specify this parameter as the    *

 *number of cmd_lines in this section plus 1. Minimum value   *

*1s 15.                                                       *
```

```
*RESTART_LIMIT= Number of attempts to restart an attached   *
*process that has terminated.                               *
*VERIFY_INTERNAL = Interval in minutes that attached        *
*processes are verified that they are still running. If this*
 *value is 0, no verification will occur. The default value *
*VERIFY-INTERVAL if not specified is 1 minute.              *
*-----------------------------------------------------------*
[MGR_ATTACH_LIST]
ATTACH_LIST_SLOTS = 15
RESTART_LIMIT = 7
VERIFY_INTERVAL = 5
CMD_LINE=(zutilmgr) RESTART=YES
CMD_LINE=(zrexxmgr) RESTART=YES
CMD_LINE=(zsnmpmgr) RESTART=YES
CMD_LINE=(zsmtrmgr) RESTART=YES
CMD_LINE=(zsmtsmgr) RESTART=YES
CMD_LINE=(zsipxmgr) RESTART=YES
CMD_LINE=(znfytmgr) RESTART=YES
CMD_LINE=(ztcpmgr)  RESTART=YES
CMD_LINE=(znetbmgr) RESTART=YES
[MGR_ATTACH_LIST]
CMD_LINE=(ztcpmgr) RESTART=YES
CMD_LINE=(zrexxmgr) RESTART=YES
CMD_LINE=(znfytmgr) RESTART=YES
ATTACH_LIST_SLOTS = 7
RESTART_LIMIT     = 7
VERIFY_INTERNAL   = 5
```

Setting Name	Value as Installed	Default Value	Minimum Value	Maximum Value
ATTACH_LIST_SLOTS	715	15Number of lines in this section	Number of lines in this section plus 115	System dependent?
RESTART_LIMIT	7	7	0 = No Restart	?3200

Setting Name	Value as Installed	Default Value	Minimum Value	Maximum Value
VERIFY_INTERVAL	5	0 = No verification	0 = No verification	?3200
CMD_LINE =	Determined by installation optionsBlank?	Determined by installation optionsBlank?	N/A	N/A
RESTART =	YES	NO	N/A	N/A

Performance Considerations

The following are the performance considerations associated with the MGR_ATTACH section:

- If the ATTACH_LIST_SLOTS value is too low, Manager tasks may be delayed while waiting in the queue for the next available slot. A value that is too high could degrade overall system performance by unnecessarily setting aside resources that remain unused.
- The RESTART_LIMIT value should be set higher when critical Manager functions are being performed. Note that regardless of the value in RESTART_LIMIT, the task will not be reinitiated if the RESTART = value is NO.
- VERIFY_INTERVAL should be set lower when critical Manager functions are being performed to ensure that vital processes are still running. A higher setting, however, might save CPU cycles when total demand is a critical factor.
- If the ZTCPMGR task is not specified, no TCP/IP communications will be enabled.
- If the ZREXXMGR task is not specified, certain EDM system methods will not be invoked. This could limit overall EDM processing capabilities.system ability.
- If the ZNFYTMGR task is not specified, you will not be able to perform Manager-initiated EDM Client Connects.

MGR_CACHE

The MGR_CACHE section of the EDMPROF.DAT file specifies cache processing options, such as cache segments, size, and statistics.

The MGR_CACHE section has the following fiveour settings:

Setting Name	Description
CACHE_SEGMENTS	Number of Cache Segments.
CACHE_SIZE	Size of each segment.
CACHE_STATS	YES|NO switch to accumulate statistics.
AVERAGE_OBJECT_SIZE	Average size of an object that will be cached.
NUM_OF_CACHE_ENTRIES	Number of entries that will be cached.

Example

```
*-MANAGER_CACHE SECTION-------------------------------------------------*

* CACHE                                                                 *

* CACHE_SEGMENTS       = The NUMBER OF CACHE SEGMENTS.                  *

*                        SET THIS VALUE TO 0 (ZERO) TO                  *

*                        TURN OFF CACHING.                              *

* CACHE_SIZE           = The SIZE OF EACH SEGMENT IN BYTES.            **

*                        Set to a installation default of               *

*                        5242880 to be able to accommodate a            *

*                             16 Meg system  may be increased           *

*                             on Real Memory availibilty                *

* CACHE_STATS           = = YES|NO SWITCH TO ACCUMULATE                 *

*                        STATISTICS. DEFAULT IS NO IF NOT               *

*                        PRESENT.                                       *

* AVERAGE_OBJECT_SIZE  = AVERAGE SIZE OF AN OBJECT THAT WILL            *

*                        BE CACHED. DEFAULT IS 2048 IF NOT              *

*                        PRESENT.                                       *

*-----------------------------------------------------------------------*

[MGR_CACHE]

CACHE_SEGMENTS = 02

CACHE_SIZE = 52428800

CACHE_STATS = NOYES
```

Setting Name	Value as Installed	Default Value	Minimum Value	Maximum Value
CACHE_SEGMENTS	2	0 = No caching	0 = No caching	System resource dependent?
CACHE_SIZE	10485752428806 Bytes	0	0	System resource dependent?
CACHE_STATS	NO	NO	N/A	N/A
AVERAGE_OBJECT_SIZE	2048?	2048 Bytes	2048?	6144?

Performance Considerations

The following are the performance considerations associated with the MGR_CACHE section:

- The settings in the MGR_CACHE section should be established based on your specific operating environment and performance needs.
- You can monitor caching performance by setting CACHE_STATS to YES.
- If your AVERAGE_OBJECT_SIZE setting is too low, the EDM Manager will have to reconfigure the cache until the object is accommodated. If the AVERAGE_OBJECT_SIZE setting is too high, you may not be making effective use of EDM's caching function.
- When modifying either of the two CACHE parameters in this section, care must be taken to be certain that you do not exceed the amount of memory available.

MGR_CLASS

The MGR_CLASS section of the EDMPROF.DAT file specifies which classes and instances will be cached during the initialization process.

The MGR_CLASS section has the following setting:

Setting Name	Description
CLASS	Specifies which classes and instances will be cached at initialization. Format: Domain.Class={Cache Class & Base Instance}, {Cache All Instances}, {Heap Size}, {Maximum Number of heaps).

Example

```
*-------------------------------------------------------------------*

* CLASS                                                             *

*  Specifies which classes/instances will be cached at initialization.*

*  Format:                                                          *

*  Domain.Class={Cache Class & Base Instance },{Cache All           *

*               Instances}, {Heap Size},{Maximum Number of          *

*               Heaps}                                              *

*-------------------------------------------------------------------*

[MGR_CLASS]

SYSTEMX.COUNTRY   = Y,Y,2048,2

SYSTEMX.DEPT      = Y,Y,2048,2

SYSTEMX.USER      = Y,Y,2048,2

SYSTEMX.WORKGRP   = Y,Y,2048,2

SYSTEMX.ZAUDITR   = Y,Y,2048,50

SYSTEMX.ZAUDITS   = Y,Y,2048,10

SYSTEMX.ZBASE     = Y,Y,2048,2

SYSTEMX.ZCID      = Y,Y,2048,1

SYSTEMX.ZLINK     = Y,Y,2048,1
```

```
SYSTEMX.ZLOCCLNT  = Y,Y,2048,1

SYSTEMX.ZLOCMGR   = Y,Y,2048,1

SYSTEMX.ZLOCNET   = Y,Y,2048,1

SYSTEMX.ZLOCSTG   = Y,Y,2048,2

SYSTEMX.ZMESSAGE  = Y,Y,2048,1

SYSTEMX.ZMETHOD   = Y,Y,2048,2

SYSTEMX.ZPAKPATH  = Y,Y,2048,1

SYSTEMX.ZPERMISS  = Y,Y,2048,1

SYSTEMX.ZOS2FLDR  = Y,Y,2048,2

SYSTEMX.ZOS2ICON  = Y,Y,2048,2

SYSTEMX.ZPACKAGE  = Y,Y,2048,2

SYSTEMX.ZSCHEDUL  = Y,Y,2048,2

SYSTEMX.ZSERVICE  = Y,Y,2048,20

SYSTEMX.ZTIME     = Y,Y,2048,2

SYSTEMX.ZWINICON  = Y,Y,2048,1

SYSTEMX.ZRSOURCE  = Y,Y,2048,6000
```

Settings

Setting Name	Description
CLASS	1. The First parameter specifies which classes/instances will be cached at initialization (Y = Yes, N = No) 2. The Second parameter specifies the initial memory allocation for instances resolved in the class during the object resolution process. The initial memory allocation is specified as the first 2 parameters combined. The third and forth parameters are the Instance Size and the Number of Instances respectively. These are only used for initial allocation, the instance size and the number of instances can be increased by the manager depending on the resolution process and the machine resources The Default Instance Size = 2048 bytes. The Default Number of Instances = 1010. **Format:** Domain.Class={Cache Class & Base Instance},{Cache All Instances},{Instance Size},{Number of Instances} The first 2 parameters after the "=" sign are used for caching and the next two are used for setting the initial parameters for building instorage instances of the class

Example

```
* MGR_CLASS SECTION                                                *
*  1. Specifies which classes/instances will be cached at         *
*     initialization (Y - Yes , N - No )                          *
*  2. Specifies the initial memory allocation for instances       *
*     resolved in the class during the object resolution          *
*      process. The initial memory allocation is specified as 2    *
*      parameters, Instance Size and the number of Instances       *
*     These are only used for initial allocation, the instance size *
*     and the number of instances can be increased by the manager  *
*     depending on the resolution process and the machine resources *
*     The Default Instance Size       = 2048 bytes                 *
*     The Default Number of Instances = 10                         *
*                                                                  *
* Format:                                                          *
* Domain.Class={Cache Class & Base Instance },{Cache All IInstances},*
*              {Instance Size},{Number of Instances}                  *
*                                                                  *
* NOTE:The first 2 parameters after the "=" sign are used for      *
* caching and the next two are used for setting the initial        *
* parameters for building instorage instances of the class         *
*------------------------------------------------------------------*
[MGR_CLASS]
SYSTEMX.ZRSOURCE  = Y,Y,4096,2000
SYSTEMX.ZSERVICE  = Y,Y,3072,20
SYSTEMX.COUNTRY   = Y,Y,2048,1
SYSTEMX.DEPT      = Y,Y,2048,1
SYSTEMX.USER      = Y,Y,2048,1
SYSTEMX.WORKGRP   = Y,Y,2048,1
SYSTEMX.ZAUDITR   = Y,Y,2048,50
SYSTEMX.ZAUDITS   = Y,Y,2048,10
```

```
SYSTEMX.ZBASE     = Y,Y,2048,1

ZACCESS.ZBASE     = Y,Y,2048,1

ZSYSTEM.ZBASE     = Y,Y,2048,1

SYSTEMX.ZCID      = Y,Y,2048,1

SYSTEMX.ZLINK     = Y,Y,2048,1

SYSTEMX.ZLOCCLNT  = Y,Y,2048,1

SYSTEMX.ZLOCMGR   = Y,Y,2048,1

SYSTEMX.ZLOCNET   = Y,Y,2048,1

SYSTEMX.ZLOCSTG   = Y,Y,2048,1

SYSTEMX.ZMESSAGE  = Y,Y,2048,1

SYSTEMX.ZMETHOD   = Y,Y,2048,1

SYSTEMX.ZPAKPATH  = Y,Y,2048,1

SYSTEMX.ZPERMISS  = Y,Y,2048,1

SYSTEMX.ZOS2FLDR  = Y,Y,2048,1

SYSTEMX.ZOS2ICON  = Y,Y,2048,1

SYSTEMX.ZPACKAGE  = Y,Y,2048,1

SYSTEMX.ZSCHEDUL  = Y,Y,2048,1

SYSTEMX.ZTIME     = Y,Y,2048,1

SYSTEMX.ZWINICON  = Y,Y,2048,1

ZACCESS.ZBASE     = Y,Y,2048,1
```

Setting Name	Value as Installed	Default Value
CLASS	See example	See example

Performance Considerations

The following are the performance considerations associated with the MGR_CLASS section:

- You can also specify your own unique classes in this section.

MGR_DIRECTORIES

The MGR_DIRECTORIES section of the EDMPROF.DAT file specifies the path for EDM's data bases, REXX methods, and non-REXX methods. See MGR_LOG to specify the directory path for the EDM Manager log.

The MGR_DIRECTORIES section has the following three settings:

Setting Name	Description
DBPATH	Fully qualified directory path for EDM Object Databases. An NTFS drive is required. Path name should be terminated with a \.
REXX_PATH	Fully qualified directory path for EDM REXX Methods. Path name should be terminated with a \.
METHOD_PATH	Fully qualified directory path for EDM Non-REXX Methods. Path name should be terminated with a \.

Example

```
*-------------------------------------------------------------------*

* MGR_DIRECTORIES SECTION                                            *

*  DBPATH         = The Path for EDM Object Databases.               *

*  REXX_PATH      = The Path for EDM REXX Methods.                   *

*  METHOD_PATH    = The Path for EDM Non-REXX Methods.               *

*-------------------------------------------------------------------*


[MGR_DIRECTORIES]

DBPATH=f:\testdb\EDMMEDMDBGR

REXX_PATH=E:\EDMMGR\REXX\

METHOD_PATH=E:\EDMMGR\BIN\
```

Setting Name	Value as Installed	Default Value
DBPATH	As specified during installation	Current DirectoryEDMMGR
REXX_PATH	EDMMGR	Current DirectoryEDMMGR
METHOD_PATH	EDMMGR	Current DirtectoryNone

While the EDMMGR is implicit here in the table above, the implication concerns its functioning with regard to the installation directory.

Performance Considerations

The following are the performance considerations associated with the MGR_DIRECTORIES section:

- The REXX directory specified in the above section is further defined by the SAMPLES subdirectory. This subdirectory contains a selected set of sample EDM REXX methods.(EDM.REXX methodology is covered within this current version of the documentation).

MGR_LICENSE

The MGR_LICENSE section of the EDMPROF.DAT file contains encrypted character-string data supplied by NOVADIGM to enforce the software licensing of EDM.

An additional feature has been added to the current version of the product whichthat enables the user to view the problems which may occur as evidenced by a corresponding E-Mail Message.

The MGR_LICENSE section has the following setting:

Setting Name	Description
LICENSE_STRING	Specifies your specific software license. Format: 16-character alphanumeric strings.
EMAIL_WARNINGS_TO	E-mail address of the system administrator - to be notified in case a problem occurs during the license verification process. The message sent out will be exactly the same message as the one written to the manager log.

Example

```
[MGR_LICENSE SECTION]

LICENSE_STRING = F904A1F206E18EE8 8A08A2D1A2E7D840 ED5894682C7C6FB1 9F152F16B13F1FCC

EMAIL_WARNINGS_TO = nobody@novadigm.com


[MGR_LICENSE]

LICENSE_STRING = 72BABED745728A84 72BABED745728A84 FBE7E22E412EC837 FBE7E22E412EC837
```

Setting Name	Value as Installed	Default Value
LICENSE_STRING	Your unique license number	N/A
EMAIL_WARNINGS_TO	N/A	N/A

Warning: Do not change this the MGR_LICENSE section unless a member of the NOVADIGM Customer Support group instructs you to do so.

Performance Considerations

The following are the performance considerations associated with the MGR_LICENSE section:

- This is essentially the Identification numbere that applies to your configuration as it cuurrently exists. If the configuration is changed then the licensing string will change. string contained in the MGR_LICENSE section for this Beta version of the EDM Manager for Windows NT is valid for 30 days only. A new string will be issued to you at that time if necessary. In addition, the Beta License string is valid for one EDM Manager and twenty EDM Clients.

MGR_LOG

The MGR_LOG section of the EDMPROF.DAT file specifies the logging directory and logging options for the EDM Manager logging facility.

The MGR_LOG section has the following fiveseven settings:

Setting Name	Description
DIRECTORY	Fully qualified directory path where the EDM Log is written
FLUSH_SIZE	Buffer size (in bytes) that the EDM logger uses to accumulate log messages before writing to the EDM:Manager log file.
THRESHHOLD	ThreshholdThreshold number of log messages that will be written to a log before automatically switching to the next log. When limit is reached, new log files are created. Specify a negative number to overwrite log file when limit is reached.
PIPE_SIZE	Maximum memory size (in bytes) of log messages processed before logged.
MESSAGE_WIDTH	The maximum width in characters of the messages in the EDM Log.
DISABLE_NT_EVENT _LOGGING = YES or NO	If YES is code, messages sent to the EDM log may be sent to the NT Event Log if they are considered critical. If NO is coded, such messages are not echoed to the NT Event Log. Note that this parameter affects only the event logging of EDM log messages. Some NT event log records are written without any corresponding EDM log messages.
DISABLE_SNMP_TRAP_LOGGIN G = YES or NO	If YES is code, messages sent to the EDM log may be sent to the primary SNMP manager as traps if they are considered critical. If NO is coded, such messages are not sent to the SNMP manager as traps. Note that this parameter affects only the trapping of EDM log messages. Some SNMP traps are issued without any corresponding EDM log messages.

The EDM: Manager is implicit within the table shown above as it functions in conjuction with the EDM Manager installation directory.

Example

```
*-------------------------------------------------------------------*
* LOG                                                               *
*  DIRECTORY      = Directory used to write the EDM Log.            *
*  FLUSH_SIZE     = Buffer size in bytes that EDM:logger uses to    *
*                   accumulate log messages before writing to the   *
*                   EDM Manager log file. A bigger buffer size      *
*                   will increase performance, but cause a delay    *
*                   in message flushing to the log file.            *
*  THRESHHOLD     = Threshold number of log messages that will be   *
*                   written to a log before automatically switching *
*                   to the next log, and so forth.                  *
```

```
*  COLUMN_WIDTH   = The width of the messages in the EDM Log.       *
*-----------------------------------------------------------------*
* NGR_LOG SECTION                                                 *
*  DIRECTORY      = Directory used to write the EDM Log.           *
*  FLUSH_SIZE     = Buffer size in bytes that EDM:logger uses to   *
*                   accumulate log messages before writing to the  *
*                   EDM:Manager log file. A bigger buffer size     *
*                   will increase performance, but cause a delay   *
*                   in message flushing to the log file.           *
*  THRESHHOLD     = Threshold number of log messages that will be  *
*                   written to a log before automatically switching *
*                   to the next log, if it is a positive number.   *
*                   If set to a negative number the log file will  *
*                   wrap around when the limit is reached.         *
*  MESSAGE_WIDTH  = The width of the messages in the EDM Manager   *
*                   Log file.                                      *
*  PIPE_SIZE      = The number of bytes to be written to a pipe    *
*                   until the messages are written to the log file *
*-----------------------------------------------------------------*

[MGR_LOG]
DIRECTORY=e:\EDMMGR\LOG
FLUSH_SIZE = 1000
THRESHHOLD = -5000000
MESSAGE_WIDTH = 256
PIPE_SIZE  = 1000000
[MGR_LOG]
DIRECTORY=E:\EDMMGR\LOG\
FLUSH_SIZE = 1000
THRESHHOLD = -500000
PIPE_SIZE  = 100000
COLUMN_WIDTH = 90
```

Setting Name	Value as Installed	Default Value	Minimum Value
DIRECTORY	EDMMGR\LOG\	Current directoryEDMMGR\LOG\	N/A
FLUSH_SIZE	1000 Bytes	5000 Bytes	?1
THRESHHOLD	-5000000 Bytes	100000 Bytes	1?
PIPE_SIZE	100000000000 Bytes	6553516KBytes	1?
MESSAGE_WIDTH	25690	90	?80

Performance Considerations

The following are the performance considerations associated with the MGR_LOG section:

- Increasing the buffer size (FLUSH_SIZE) will enhance performance, but will delay messages flushed to the log file.
- Increase MESSAGE_WIDTH if log messages are being truncated.
- If you are closely monitoring system status using the Manager log, set the THRESHHOLD value to a positive value to create and save successive portions of the Manager log. If storage is critical, set value to a negative value to reuse the allocated log disk space.
- If you are invoking a number of EDM methods, you can control the size of the Manager log using the MSG_LIMIT setting of the MGR_METHOD section. Additionally, the TASK_LOG_LIM controls the number of messages generated by the execution of tasks
- When modifying parameters in this section as they relate to memory or disk utilization, care must be taken to be certain you do not exceed the maximum amount of memory or storage space available..

> **Warning**: You must save your Manager log if you are running the EDM Manager for Windows NT as a service and you plan to reboot your system. If no action is taken, you will lose the contents of the Manager log for the previous Manager session as the log will be overwritten for the new Manager session.

MGR_METHODS

The MGR_METHODS section of the EDMPROF.DAT file specifies options for method execution.

The MGR_METHODS section has the following three settings:

Setting Name	Description
MSG_LIMIT	Maximum number of messages a method can issue to the EDM log. Note that this setting is invoked only when the LOG_LIMIT setting is not present.
LOG_LIMIT	Maximum number of messages a method can issue to the EDM log. When this limit is reached, a message will be written stating that the message limit has been reached and that all other messages from the method will be ignored.
TIMEOUT	Manager Method timeout parameter.

Example

```
*-------------------------------------------------------------------------*

* MGR_METHODS SECTION                                                     *

*  LOG_LIMITMSG_LIMIT(LOG_LIMIT) = Maximum number of messages a method can *

*                          issue to the EDM log. Default is 0, i.e.       *

*                          no limitation.                                 *

*  TIMEOUT              = Manager Method Timeout in seconds.              *

*-------------------------------------------------------------------------*


[MGR_METHODS]

MSG_LIMIT = 0

LOG_LIMIT = 0

TIMEOUT   = 300
```

Setting Name	Value as Installed	Default Value	Minimum Value
MSG_LIMIT	2000 Bytes	0 = No Limit	0
LOG_LIMIT	0?	0 = No Limit	
TIMEOUT	300 Seconds	60 Seconds	0

Performance Considerations

The following are the performance considerations associated with the MGR_METHODS section:

- The number of messages generated by the execution of a method is also effected by the TASK_LOG_LIM setting in the MGR_TASK_LIMIT section.
- You can tune system resource usage with the TIMEOUT setting. If system resources are critical, a lower TIMEOUT setting will free up unused processing cycles.

MGR_NOTIFY

The MGR_NOTIFY section of the EDMPROF.DAT file specifies the defaults for EDM Manager notification of EDM Clients.

The MGR_NOTIFY section has the following three settings:

Setting Name	Description
NFYT_TIMEOUT	Notify time-out (in seconds) for TCP/IP clients
NFY6_TIMEOUT	LU62 Notify time-out in minutes
NFY6_TPN	LU62 Default TPNAME for Notify

Example

```
*-------------------------------------------------------------------*

* MGR_NOTIFY SECTION                                                *

*  NFYT_TIMEOUT   = Notify time-out for TCP/IP clients.             *

*  NFY6_TIMEOUT   = LU62 Notify time-out in minutes.                *

*  NFY6_TPN       = LU62 Default TPNAME for Notify.                 *

*-------------------------------------------------------------------*

[MGR_NOTIFY]

NFYT_TIMEOUT = 12060

NFY6_TPN     = EDMINITN
```

Setting Name	Value as Installed	Default Value
NFYT_TIMEOUT	60120 Seconds	0
NFY6_TIMEOUT		0
NFY6_TPN	EDMINITN	0

Performance Considerations

The following are the performance considerations associated with the MGR_NOTIFY section:

- You should establish your MGR_NOTIFY settings based on your network operations parameters.
- The MGR_NOTIFY settings should also be coordinated with values in the MGR_RETRY section, as well as the MGR_TASK_LIMIT section.

MGR_OBJECT_RESOLUTION

The MGR_OBJECT_RESOLUTION section of the EDMPROF.DAT file specifies a method to handle the ZADMIN object. Used for compatibility with the pre-3.0A EDM data bases.

The MGR_OBJECT_RESOLUTION section has the following two settings:

Setting Name	Description
ALWAYS_CALL_ ZADMIN	Force object resolution to call to ZADMIN method to process ZADMIN object. Values are YES/NO. Used for compatibility with pre-V30A databases.
ALLOW_CIRCULAR_ REFERENCES	Allow or disallow circular references between classes in the same resolution path.

Example

```
*-----------------------------------------------------------*

* MGR_OBJECT_RESOLUTION SECTION                             *

*  ALWAYS_CALL_ZADMIN = (YES|NO) Force call to ZADMIN       *

*                        method to handle ZADMIN object.    *

*                        Used for compatibility with        *

*                        pre-V30A databases.                *

*  ALLOW_CIRCULAR_REFERENCES = YES  Default is NO if not    *

*                              present.                     *

*-----------------------------------------------------------*


[MGR_OBJECT_RESOLUTION]

ALWAYS_CALL_ZADMIN = YES
```

Setting Name	Value as Installed	Default Value
ALWAYS_CALL_ZADMIN	YES	NO
ALLOW_CIRCULAR_REFERENCES	NO	NO

Performance Considerations

The following are the performance considerations associated with the MGR_OBJECT_RESOLUTION section:

- The ALLOW_CIRCULAR_REFERENCES setting should not be set as YES unless specific conditions warrant it and provisions have been made for its operation.

MGR_RETRY

The MGR_RETRY section of the EDMPROF.DAT file specifies—in minutes—when an EDM Client can attempt another connection with the EDM Manager, after being rejected due to an exceeded task limit or a disabled connection.

The MGR_RETRY section has the following two settings:

Setting Name	Description
BUSY_RETRY	Number of minutes for EDM Client to wait before reconnecting when the EDM Manager is at task limit (TASK_LIMIT).
DISA_RETRY	Number of minutes for EDM Client to wait before reconnecting when the EDM Manager has logons halted.

Example

```
*-------------------------------------------------------------------------------------*

* MGR_RETRY SECTION                                                                    *

               *                                                                      *

*  BUSY_RETRY = Number of minutes for EDM Client to retry when EDM Manager is at task limit*

*                   (TASK_LIMIT).                          *

*  DISA_RETRY = Number of minutes for EDM Client to retry when EDM Manager has logons      *

*                  halted.                                 *

*-----------------------------------------------------------*

[MGR_RETRY]

BUSY_RETRY=1

DISA_RETRY=999
```

Setting Name	Value as Installed	Default Value	Minimum Value	Maximum Value
BUSY_RETRY	7 Minutes	0 = Allow connect now	0 = Allow connect now	0 = Allow connect now
DISA_RETRY	999 = Do not retry.	999 = Do not retry.	0 = Allow connect now	9990 = Do not retryAllow connect now

Performance Considerations

The following are the performance considerations associated with the MGR_RETRY section:

- You can raise both MGR_RETRY values if processing resources are critical. Lower values will provide greater assurance that connections will be re-established if broken during client connects.
- The MGR_RETRY settings should also be coordinated with values in the MGR_NOTIFY section, as well as the MGR_TASK_LIMIT section.

MGR_SMTP_MAIL

The MGR_SMTP_MAIL section of the EDMPROF.DAT file specifies the ID of the EDM Manager, and the TCP port number of the EDM Manager to be used.

The MGR_SMTP_MAIL section has the following five settings:

Setting Name	Description
MGR_MAIL_ID	Mail ID for the Manager. Mail Receiving Manager will reject mail addressed to any other userid.
MAIL_DIR	Directory to spool and queue outgoing mail from the SMTP Send Manager.
SMTP_PORT	Port to wait for incoming mail.
RETRY_INTERVAL	Specifies in seconds how long the Manager will wait before retrying to deliver undelivered mail from the spool.
MAX_TIME_IN_SPOOL	Specifies in minutes how long the Manager will retain spooled messages before deleting them.

Example

```
*-------------------------------------------------------------------*

* MANAGERanager SMTP MAILail SECTIONection                                    *

* This section is referred by zsmtsmgr (SMTP send Manager) and       *

* zsmtrmgr (SMTP receive Manager:                                    *

* Keywords :                                                         *

* MGR_MAIL_ID = The mailid for the manager is specified in this field*

* The Mail Receiving Manager, would accept mail addressed to this    *

* user id, mail addressed to any other user id would be rejected     *

* Default MGR_MAIL_ID is edm (Case Sensitive)                        *

* MAIL_DIR: = The directory to spool and queue out going mails       *

* from the zsmtsmgr  Default MAIL_DIR is . (current directory)       *

* SMTP_PORT: = This is the port on which the SMTP receive            *

* Manager waits for incoming mail. The Default SMTP PORT is 25       *

* which is the reserved port for SMTP. On systems where a mail       *

* server eg. sendmail is running this port might already be in       *

* use and it would conflict with the zsmtrmgr                        *

* RETRY_INTERVAL:  = This field is used to specify the time in       *

* seconds, after which undelivered mail in the spool, would be       *

* retried for delivery  Default is 300 seconds                       *

* MAX_TIME_IN_SPOOL: = This field is used to specify the time        *

* in  minutes, after which an undelivered mail in spool, would       *
```

```
* be deleted. Default is 4320 minutes (3 Days)                          *

[MGR_SMTP_MAIL]

RETRY_INTERVAL = 300

MAX_TIME_IN_SPOOL = 4320

SMTP_PORT=25

MAIL_DIR=e:\EDMMGR\MAIL

MGR_MAIL-ID=edm

The Mail Receiving Manager, would accept mail addressed to this *

* user id, mail addressed to any other user id would be rejected*

* Default MGR_MAIL_ID is edm (Case Sensitive)                      *

* MAIL_DIR: = The directory to spool and queue out going mails  *

* from the zsmtsmgr  Default MAIL_DIR is . (current directory)  *

* SMTP_PORT: = This is the port on which the SMTP receive       *

* Manager waits for incoming mail. The Default SMTP PORT is 25  *

* which is the reserved port for SMTP. On systems where a mail  *

* server e.g. sendmail is running this port might already be in *

* use and it would conflict with the zsmtrmgr                   *

* RETRY_INTERVAL:  = This field is used to specify the time in  *

* seconds, after which undelivered mail in the spool, would be  *

* retried for delivery  Default is 300 seconds                  *

* MAX_TIME_IN_SPOOL: = This field is used to specify the time   *

* in  minutes, after which an undelivered mail in spool, would  *

* be deleted. Default is 4320 minutes (3 Days)                  *

[MGR_SMTP_MAIL]

RETRY_INTERVAL = 300

MAX_TIME_IN_SPOOL = 4320
```

Setting Name	Value as Installed	Default Value
MGR_MAIL_ID	As specified during installation	edm (Case Sensitive)
MAIL_DIR	As specified during installation	Current Directory
SMTP_PORT	As specified during installation	25
RETRY_ INTERVAL	300 Seconds	300 Seconds
MAX_TIME_IN_SPOOLMGR_MAIL_ID	4320 MinutesAs specified during installation	4320 Minutes (= 3 Days)edm (Case Sensitive)

Performance Considerations

The following are the performance considerations associated with the MGR_SMTP_MAIL section:

- If storage capacity is an issue, lower the MAX_TIME_IN_SPOOL value. This will decrease the amount of time messages are retained.
- Lower RETRY_INTERVAL values will use more system processing resources

MGR_SNMP

The MGR_SNMP section of the EDMPROF.DAT file specifies how the SNMP agent built into the Manager handles SNMP transactions.

The MGR_SNMP section has the following 9 Settings

Settings

Setting Name	Description
RUN_AS_EXTENSION	This receives a Yes or a No value indicating whether or not the Manager will act as the primary SNMP agent.. If YES is coded, NT's SNMP service is the primary SNMP agent and EDM SNMP transactions are processed by an EDM SNMP extension DLL. (In this case, the SNMP_COMMUNITY, SNMP_PORT and SNMP_IP_ADDR are not used since SNMP port access is handled by NT's SNMP service) If NO is coded, the EDM:Manager will act as the primary SNMP agent. In that case SNMP_COMMUNITY parameter should be specified while SNMP_IP_ADDR and SNMP_PORT may be specified to override their default values.
COMMUNITY_NAME	This parameter should be set to a character string. The string will be used as the SNMP community name by the EDM agent. The community name is effectively a password which incoming SNMP transactions must match. A default value of "public" is used if this keyword is not specified. This keyword is only effective when RUN_AS_EXTENSION is set to NO.
SNMP_PORT	This parameter is used to specify the TCP/IP port on which the EDM agent receives SNMP transactions. If not specified, the default is port 161. This keyword is only effective when RUN_AS_EXTENSION is set to NO.
SNMP_IP_ADDR	This parameter specifies the TCP/IP address of the network adapter card on which the agent is to receive SNMP transactions. When not specified or if specified as 0.0.0.0 any adapter on the machine can be used. The default is that any adapter on the machine can be used. This keyword is only used when RUN_AS_EXTENSION is set to NO and there are several adapters on the machine and a specific adapter to is to receive SNMP transactions
SNMP_MANAGER_IP_ADDR SNMP_MANAGER_IP_ADDR2 SNMP_MANAGER_IP_ADDR3	The SNMP managers at these IP addresses are authorized to issue get and set commands for variables supported by the EDM agent. If all three of these parameters are not specified then any SNMP manager with the correct community name is authorized to run SNMP set and get commands on the EDM agent. If any of these three parameters are specified then only commands coming from those IP addresses will be processed. The SNMP manager specified by SNMP_MANAGER_IP_ADDR is considered the primary SNMP manager. The primary SNMP manager is the manager which will receive all traps generated by the EDM:Manager. If this parameter is not specified or if it is set to 0.0.0.0 then traps will not be issued by the EDM:Manager.
SNMP_SET_ COMMUNITY	This parameter may be set to a character string. The EDM agent will use this string as the SNMP community name when it is attempting to authorize SET commands. If this keyword is not specified, the community name given by SNMP_COMMUNITY is used for SET commands. This keyword is only effective when RUN_AS_EXTENSION is set to NO
SNMP_MANAGER_ PORT	This parameter is used to specify the remote TCP/IP port to which the EDM: Manager sends its traps. The default is port 162.
SNMP_LOGGER_ PORT	This parameter is used to specify the local TCP/IP port on which the EDM: Manager sends its traps. If this parameter is not specified, the EDM: Manager uses an ephemeral port, that is, a system assigned port. (*An ephemeral port works well in all case(s*)
SNMP_ZERROR_ SEVERITY	This parameter is used to specify the severity of ZERROR instances to send as SNMP traps. The trap is sent when the EDM: Manager adds an error instance to its ZERRORM for an error whose severity is greater than or equal to the value specified by this parameter. The parameter can be set to a positive value between 0 and 99, the default is 12.

Example

```
* MGR_SNMP SECTION                                                *
```

```
*  RUN_AS_EXTENSION = YES or NO                                          *
*                     YES means NT's SNMP service is the primary SNSNMP   *
*                     agent and EDM SNMP transactions are processed by an *
*                     EDM SNMP extension DLL. If set to YES               *
*                     SNMP_PORT and SNMP_IP_ADDR are not used since SNMP  *
*                     port access is handled by NT's SNMP service in this *
*                     case. NO means the EDM:Manager will act as the      *
*                     primary SNMP agent. In that case, COMMUNITY_NAME    *
*                     should be specified while SNMP_IP_ADDR and SNMP_PORT*
*                     may be specified to override their default values.
*.Example 1:                                                             *
* [MGR_SNMP]                                                             *
* NT's SNMP service will be run the SNMP agent port:                     *
* RUN_AS_EXTENSION = YES                                                 *
* Address of SNMP manager which receives traps and performs get/sets:    *
* SNMP_MANAGER_IP_ADDR = 204.7.83.99                                     *
* Send traps for ZERROR of severity 8 or more:                           *
* SNMP_ZERROR_SEVERITY = 8                                               *
* Example 2:                                                             *
* The EDM Manager agent will own the SNMP agent port:                    *
* RUN_AS_EXTENSION = NO                                                  *
* Password for gets: COMMUNITY_NAME = AG00021                            *
* Password for sets: SET_COMMUNITY_NAME = AG0550081                      *
* Address of SNMP manager which receives traps and performs get/sets:    *
* SNMP_MANAGER_IP_ADDR  = 204.7.83.99                                    *
* Address of second SNMP Manager which can also performs get/sets:       *
* SNMP_MANAGER_IP_ADDR2 = 204.7.83.206                                   *
* COMMUNITY_NAME   = public                                              *
*                     The community name is effectively a password which  *
```

```
*                     incoming SNMP transactions must match. The above    *
*                     value of "public" is also used as the default       *
*                     community name when this keyword is not specified.  *
*                     This keyword is only effective when RUN_AS_EXTENSION*
*                     is set to NO.                                       *
*  SNMP_PORT        = 161                                                 *
*                     This is the TCP/IP port on which the agent receives *
*                      SNMP transactions.  If not specified the default is*
*                      port 161. This keyword is only effective when      *
*                     RUN_AS_EXTENSION is set to NO.                      *
*  SNMP_IP_ADDR     = 0.0.0.0                                             *
*                     This specifies the TCP/IP address of the network    *
*                     adapter card on which the agent is to receive SNMP  *
*                     transactions.  When not specified, the default is   *
*                     port 0.0.0.0 which means that any adapter on the    *
*                     machine can be used.  This keyword is only used     *
*                     when RUN_AS_EXTENSION is set to NO and there are    *
*                     several adapters on the machine and a specific      *
*                     adapter to is to re-=receive SNMP transactions.     *
* SNMP_MANAGER_IP_ADDR                                                    *
* SNMP_MANAGER_IP_ADDR2                                                   *
* SNMP_MANAGER_IP_ADDR3                                                   *
* The SNMP managers at these IP addresses are authorized to issue get and *
* set commands for variables supported by the EDM agent. If all three of  *
* these parameters are not specified then any SNMP manager with the       *
* correct community name is authorized to run SNMP set and get commands on*
* the EDM agent.  If any of these three parameters are specified then only*
* commands coming from those IP addresses will be processed.              *
* The SNMP manager specified by SNMP_MANAGER_IP_ADDR is considered the    *
```

```
* primary SNMP manager. The primary SNMP manager is the manager which will*
* receive all traps generated by the* EDM:Manager. If this parameter is   *
* not specified or if it is set to 0.0.0.0 then traps will not be issued  *
* by the EDM:Manager.                                                     *
* SNMP_LOGGER_PORT                                                        *
* This parameter is used to specify the local TCP/IP port on which the    *
* EDM:Manager sends its traps. If this parameter is not specified, the    *
* EDM:Manager uses an ephemeral port, that is, a system assigned port. (An*
* ephemeral port works well in all cases).                                *
* SNMP_ZERROR_SEVERITY                                                    *
* This parameter is used to specify the severity of ZERROR instances to   *
* send as SNMP traps. The trap is sent when the EDM:Manager adds an error *
* instance to its ZERRORM for an error whose severity is greater than or  *
* equal to the value specified by this parameter. The parameter can be set*
* to a positive value between 0 and 99, the default is 12.                *
```

Setting Values

Setting Name	Value as Installed	Default Value
RUN_AS_EXTENSION	Y for Yes NT's SNMP service is primary SNMP agent	N for No EDM SNMP agent is primary SNMP agent
COMMUNITY_NAME	Public	Public
SNMP_PORT	161	161
SNMP_IP_ADDR	0.0.0.0	

Performance Considerations

There are no “performance” issues with any of the SNMP parameters. If you don’t start zsnmpmgr then you are not starting the agent and you are running one less task in the Manager.

Note: If running as an extension, you need the agent running (zsnmpmgr) as well as the extension .dll.

.MGR_STARTUP

The MGR_STARTUP section of the EDMPROF.DAT file specifies the ID of the EDM Manager, and the TCP port number of the EDM Manager

> **Note:** (Do not change MGR_ID, MANAGER_TYPE,MGR_NAME after a successful install unless instructed to do so by Novadigm Technical Services)

The MGR_STARTUP section has the following seven settings:

Setting Name	Description
MGR_ID	Qualifier for EDM log file. **Format**: 3 character hexadecimal. This ID is also passed to the EDM Client as ZOBJMID variable, and used in the data base to identify the source manager in a Distributed Manager environment.
MANAGER_TYPE	Type of Manager. . Values are Distributed, Server, or Standalone.
MGR_NAME	Manager name.
TCP_PORT	Port to listen on for TCP/IP EDM Client connections. Must be greater than or equal to 1029.
NETB_NAME	NetBIOS name. Must be a valid NetBIOS name.
NETB_LANANUM	NetBIOS LAN adapter number. Must be a valid NetBIOS LAN adapter number.Values supported are) to 256.
SIPX_SOCKET	Socket to listen on for Novell SPX/IPX EDM Client connections.
ALLOW_DUPLICATE_ IP_ADDRESSSIPX_SOCKET	Values are Yes or No. YesA setting of NO will cause the manager to reject a second login (edmdemon session) with the EDM Manager when one is already active from the same IP address. No will A setting of YES will allowmallow multiple concurrent IP connections from the same Client IP AddressSocket to listen on for Novell SPX/IPX EDM Client connections.

Example

```
* STARTUP SECTION                                                              *
*  MANAGER TYPE   = Type of EDM:Manager installed. Installation program adds this setting. *
*  MGR_ID         = Qualifier for EDM log file. This ID is also passed to the EDM:Client as*
*                   ZOBJMID variable.                                          *
*  MANAGER_NAME   = Manager Name specified at installation. Do not change MANAGER_TYPE,    *
*                   MGR_ID or MGR_NAME after the EDM:MANAGER is installed without being    *
*                   advised to do so by TECHNICAL SERVICES                     *
*  TCP_PORT       = Port to listen on for EDM:Client connections.             *
*  SHOW_VERINFO   = YES or NO                                                 *
*                   YES will show Manager module version information to Manager log file.  *
*                   NO will not report module versions.                        *
*-----------------------------------------------------------------------------*
[MGR_STARTUP]
MANAGER_TYPE = DISTRIBUTED
```

```
MGR_ID=001

MGR_NAME = EDM

SHOW_VERINFO = YES

SIPX_SOCKET=3460

TCP_PORT=3460

NETB_LANANUM=0

NETB_NAME=EDMNET

*-------------------------------------------------------------------------------*

* STARTUP                                                                       *

*  MGR_ID       = Qualifier for EDM log file. This ID is also passed to the EDM   *

*                 Client as ZOBJMID variable.                                   *

*  TCP_PORT     = Port to listen on for EDM Client connections.                  *

*-------------------------------------------------------------------------------*

[MGR_STARTUP]

MGR_ID   = 007

TCP_PORT = 1029

NETB_NAME = EDMNET

NETB_LANANUM = 0
```

Setting Name	Value as Installed	Default Value	Minimum Value	Maximum Value
MGR_ID	As specified during installation	001	000	FFF
MANAGER_ TYPE	As specified during installation	DISTRIBUTEDStandalone	N/A	N/A
MGR_NAME	As specified during installation	EDM_MANAGER_001	N/A	N/A
TCP_PORT	As specified during iInstallation default is 3460 unless otherwise specified during installation	1029	System dependent	System dependent
NETB_NAME	As specified during installation	EDMNET	N/A	N/A
NETB_ LANANUM	As specified during installation	0	0	256
SIPX_SOCKET	Installation default is 3460 unless otherwise specified during installation	1029	System dependent	System dependent
ALLOW_ DUPLICATE_IP_ADD RESSSIPX_SOCKET	YESAs specified during installation	YES1029	N/A	N/A

Performance Considerations

The following are the performance considerations associated with the MGR_STARTUP section:

- If your MGR_TYPE is not set for Distributed, you will not be able to use Distributed Manager Adapter functionality.
- The MGR_ID value must be a three character hexadecimal string.

MGR_TASK_LIMIT

The MGR_TASK_LIMIT section of the EDMPROF.DAT file specifies the maximum number of the following:

- Concurrent tasks allowed.
- Concurrent tasks allowed for EDM Clients that can connect later.
- Lines that can be written to the EDM Manager log per EDM Client.
- Resolutions allowed per EDM Client.

The MGR_TASK_LIMIT section has the following settings:

Setting Name	Description
TASKLIM_HARD (TASKLIM)	The maximum number of concurrent tasks for the EDM Manager . . These tasks include all EDM Manager system tasks, tasks specified in ATTACH_LIST, and all the tasks created as a result of the connect process initiated by EDM Clients and EDM Administrators. No connection to the EDM Manager is accepted once this limit is reached. You can specify either TASKLIM_HARD or TASKLIM in the profile. If both are specified, then TASKLIM_HARD will be used.
TASK_LOG_LIM	The maximum number of lines that can be written to the EDM Manager log per EDM Client.
TASK_RESO_LIMIT	The maximum number of resolutions allowed per EDM Client.
TASK_STACK_SIZE	This setting defines how many variables may be used per task.

Example

```
* TASK_LIMIT SECTION                                                              *
*   TASKLIM        = Maximum number of EDM:Manager tasks (MIN=15)                 *
*   TASK_LOG_LIM   = Maximum number of lines that can be written to the log file for one   *
*                    process(EDM:Client).  The default is 0 (no limit).          *
*   TASK_RESO_LIM  = Maximum number of resolutions allowed for one EDM:Client. The default *
*                    is 64,000.                                                  *
*   TASK_STACK_SIZE= Stack size for processing.                                   *
*-------------------------------------------------------------------------------*
[MGR_TASK_LIMIT]
TASKLIM  = 20
TASK_LOG_LIM    = 0
TASK_RESO_LIM   = 64000
TASK_STACK_SIZE = 64000
*-------------------------------------------------------------------------------*
*  TASK_LIMIT                                                                   *
```

```
*  TASKLIM_HARD   = Maximum number of EDM Manager tasks (the minimum is 25)       *
*  TASKLIM_SOFT   = When this number is reached, the EDM Manager accepts          *
*                   connections only for EDM Clients that cannot retry later.    *
*                   However, as TASKLIM is reached, all other requests will be   *
*                   refused, regardless of the EDM Clients' ability to reconnect. *
*  TASK_LOG_LIM    = Maximum number of lines that can be written to the log file  *
*                   for one process (EDM Client).  The default is 100,000.       *
*  TASK_RESO_LIM   = Maximum number of resolutions allowed for one EDM Client. The*
*                     default is 64,000.                                         *
*-------------------------------------------------------------------------------*

[MGR_TASK_LIMIT]

TASKLIM_HARD=50

TASKLIM_SOFT=45

TASK_LOG_LIM=100000

TASK_RESO_LIM=64000
```

Setting Name	Value as Installed	Default Value	Minimum Value	Maximum Value
TASK_LOG_LIM	0?	100000 Lines	?0	N/A?
TASK_RESO_LIMIT	64000?	64000 Resolutions	1?	?429496729564000
TASK_STACK_SIZE	64000?	6400064k	16384?	640004294967295?

Performance Considerations

The following are the performance considerations associated with the MGR_TASK_LIMIT section:

- We recommend that the difference between the TASKLIM_HARD and the TASKLIM_SOFT should be 10% of the TASKLIM_HARD. For example, if the TASKLIM_HARD is 50, then the difference between the TASKLIM_HARD and the TASKLIM_SOFT should be 10% of 50, or 5. Thus, the TASKLIM_SOFT is 50 minus 5, or 45.

- Using the illustration above, that is if TASKLIM_SOFT is 45 and TASKLIM_HARD is 50, then the EDM Manager will accept connections for EDM Clients, whether or not they can connect later, until reaching the 45 task soft limit. After reaching that limit, EDM Manager will only accept connections for EDM Clients that *cannot* connect later, until reaching the 50 task hard limit. After reaching that limit, EDM Manager will not accept any connections.

- The MGR_TASK_LIMIT settings should also be coordinated with values in the MGR_NOTIFY section, as well as the MGR_RETRY section.

MGR_TIMEOUT

The MGR_TIMEOUT section of the EDMPROF.DAT file specifies, in seconds, how long the EDM Manager will wait for a request from a connected EDM Client before disconnecting that EDM Client due to inactivity (no requests/responses from the EDM Client).

The MGR_TIMEOUT section has the following three two settings:

Setting Name	Description
TIMEOUT_COMM	Communications (receive) time-out in seconds.
ADMIN_TIMEOUT	Time-out in seconds for EDM Administrator Browser functions.

Example

```
*-------------------------------------------------------------------------------------*

* TIMEOUT SECTION                                                                     *

*  TIMEOUT_COMM  = Communications (receive) time-out in seconds.                      *

*  ADMIN_TIMEOUT  = Timeout for administrator in seconds. However, a setting of 0 indicates*

*                   no timeout for EDM:Explorer and EDM:Administrator transactions.   *

*-------------------------------------------------------------------------------------*

[MGR_TIMEOUT]

TIMEOUT_COMM  = 31800

ADMIN_TIMEOUT = 0

* TIMEOUT (in seconds)    ------------------------------                              *

*  TIMEOUT_COMM   = Communications (receive) time-out                                 *

*  TIMEOUT_NCOMM  = Non-communications time-out                                       *

*-------------------------------------------------------------------------------------*

[MGR_TIMEOUT]

TIMEOUT_COMM  = 1800

TIMEOUT_NCOMM = 300
```

Setting Name	Value as Installed	Default Value	Minimum Value	Maximum Value
TIMEOUT_COMM	31800000 Seconds=Never Time-out	0=Never Time-out	0	?42949672953200
ADMIN_TIMEOUT	0=Never Time-out	0=Never Time-out	0	4294967295N/A

Performance Considerations

The following are the performance considerations associated with the MGR_TIMEOUT section:

- You can lower both MGR_TIMEOUT values if processing resources are critical.
- The MGR_TIMEOUT settings should also be coordinated with values in the MGR_RETRY section.

MGR_TPINIT

The MGR_TPINIT section of the EDMPROF.DAT file specifies packet sizes to send to EDM Clients.

The MGR_TPINIT section has the following six settings:

Setting Name	Description
BUFTCP	TCP Buffer size used for Send/Receive.
BUFLU2	LU2 Buffer size used for Send/Receive.
BUFLU62	LU6.2 Buffer size used for Send/Receive.
BUFNETB	NetBIOS Buffer size used for Send/Receive.
BUFSIPX	SPX/IPX Buffer size used for Send/Receive.
MAXREC	Maximum record size.

Warning: Do not change any setting in this section unless a member of the NOVADIGM Customer Support group instructs you to do so.

Example

```
*-------------------------------------------------------*
-*TPINIT  SECTION                                       *
*  BUFTCP   = TCP Buffer size (bytes) used for Send/Receive.*
*  BUFNETB  = NetBIOS Buffer size (bytes) used for      *
*             Send/Receive                              *
*  BUFLU62  = LU6.2 Buffer size (bytes)used for         *
*             Send/Receive.                             *
*  BUFLU2   = LU2 Buffer size (bytes)used for Send/Receive. *
*  BUFSIPX  = IPX/SPX Buffer (bytes) size used for      *
*             Send/Receive.                             *
*  MAXREC   = Maximum record size.                      *
*  DO NOT CHANGE ANY OF THE PARAMETERS IN THE MGR_TPINIT    *
*  SECTION UNLESS ADVISED TO DO SO BY NOVADIGM TECHNICAL    *
*  SERVICES                                             *
[MGR_TPINIT]
BUFTCP  = 12288
BUFNETB =  4096
BUFLU2  =   900
BUFLU62 = 12288
```

```
BUFSIPX = 10240

MAXREC  =  6144

* BUFTCP        = TCP Buffer size used for Send/Receive.    *

* BUFNETB       = NETBIOS Buffer size used for Send/Receive.*

* Default value is 4096.                                     *

* MAXREC        = Maximum record size. ***Do not change.  ***

*-------------------------------------------------------------*

[MGR_TPINIT]

BUFTCP  = 12288

MAXREC  = 6144
```

Setting Name	Value as Installed	Default Value
BUFTCP	12288	12288
BUFLU2	900	4096
BUFLU62	1228	12288
BUFNETB	4096	4096
BUFSIPX	10240	4096
MAXREC	6144	6144

Performance Considerations

The following are the performance considerations associated with the MGR_TPINIT section:

The buffer size settings in the MGR_TPINIT section should only be changed in coordination with equivalent changes to EDM Clients.

MGR_TRACE

The MGR_TRACE section of the EDMPROF.DAT file controls and influences diagnostic logging for the EDM Manager for Windows NT. This section contains a list of keywords that you can specify.

All diagnostic output produced by TRACE keywords is written to the active EDM activity log. To activate a TRACE keyword, type `YES`. To deselect a TRACE keyword, type `NO`.

TRACE keywords specified in the MGR_TRACE section are invoked at EDM Manager initialization and remain in effect until changed by altering the MGR_TRACE setting and restarting the EDM Manager. The trace settings in effect at EDM Manager initialization are displayed at the beginning of EDM activity log.

The following table, The EDM Manager TRACE Keyword List for Windows NT, provides a list of each valid TRACE keyword, and a brief description.

The EDM Manager TRACE Keyword List for Windows NT

TRACE Keywords	Definitions
ADMIN	Traces ADMIN transaction flow.
ADMPROM	Not used.
ALL	Turns on all other traces.
ALLOC	Traces file allocations.
AUDIT	Traces audit file activity.
BUFF	Traces data buffers (without transformation).
CMPR	Traces data compression.
COMM	Traces 3270 data stream buffers.
COMMDATA	Traces data communications.
COMMRPLS	Traces Communications Control Blocks (CCBS).
CONFIG	Traces configuration file activities.
CPIC	Traces LU6.2 protocol flows. Not supported by EDM Manager for Windows NT.
DATA	Traces data buffers to or from the EDM Client.
DSCOMP	Traces 3270 data stream compression. MVS only.
ENQDEQ	Traces serialization activity (Enqueues/dequeues).
EXPL	Traces data transformation (explode).
FILE	Traces file I/O.
IMPL	Traces data transformation (implode).
LOOKASID	Traces cache activity for classes/instances.
METHOD	Traces EDM Manager method execution/return codes.
NOTIFY	Traces notify processing.
OBJCRC	Traces object CRC processing.
OBJRES	Traces object resolution (very detailed).
OBJRES1	Traces object resolution (medium detail).
OBJRESO	Traces high level object resolution flow (light detail).
OBJXFER	Traces object transfer.
PASSWORD	Traces passwords.
PROFILE	Traces profile data base activity.
PROMOTE	Traces file promotion.
RESOURCE	Traces resource file activity.

TRACE Keywords	Definitions
REXX	Traces REXX environment.
REXXOFF	Suppresses all REXX activity.
STATS	Traces statistics.
SUBST	Traces variable substitution.
TCP	Traces TCP/IP activity.
TEST	-Reserved-
TRAN	Traces data transformation buffers. MVS only.
VAR	Traces the variable references.
VARSTG	Traces variable processing storage usage.
VSAM	Traces VSAM I/O. MVS only.
VSAMDATA	Traces VSAM data. MVS only.
VSAMRPLS	Traces VSAM Request Parameter List (RPL). MVS only.
YEAR2000	Traces a database's Year 2000 compliance

Example

```
*-------------------------------------------------------------------------------------------*

* TRACE SECTION                                                                            *

*  Refer to The EDM Manager Operations Guide for details about specifying the TRACE flags  *

*-------------------------------------------------------------------------------------------*

[MGR_TRACE]

ADMIN    = NO

ADMPROM  = NO

ALL      = NO

ALLOC    = NO

AUDIT    = NO

BUFF     = NO

CMPR     = NO

COMM     = NO

COMMDATA = NO

COMMRPLS = NO

CONFIG   = NO

CPIC     = NO

DATA     = NO

DSCOMP   = NO

ENQDEQ   = NO

EXPL     = NO
```

```
FILE     = NO
IMPL     = NO
LOOKASID = NO
METHOD   = NO
NOTIFY   = NO
OBJCRC   = NO
OBJRES   = NO
OBJRES0  = NO
OBJRES1  = NO
OBJXFER  = NO
PASSWORD = NO
PROFILE  = NO
PROMOTE  = NO
RESOURCE = NO
REXX     = NO
REXXOFF  = NO
STATS    = NO
SUBST    = NO
[MGR_TRACE]
TCP      = NO
TEST     = NO
TRAN     = NO
VAR      = NO
VARSTG   = NO
VSAM     = NO
VSAMDATA = NO
VSAMRPLS = NO
YEAR2000 = NO
```

Performance Considerations

The following are the performance considerations associated with the MGR_TRACE section:

- Turning on trace flags generates a large number of Manager log messages. This will degrade the performance of the EDM Manager due to the disk I/O load. However, this may be necessary at times for problem resolution. Ensure that your Manager log is properly configured (MGR_LOG).

MGR_USERLOG

The MGR_USERLOG section of the EDMPROF.DAT file specifies the logging directory and logging options for the EDM User logging facility.

The MGR_LOG section has the following six settings:

Setting Name	Description
ACTIVATE	Activate User Log at Manager Start-up. Values for this setting are Yes or No.
DIRECTORY	Fully qualified directory path where the EDM Log is written. Path name should be terminated with a \.
THRESHHOLD	Threshold number of log messages that will be written to a log before automatically switching to the next log. When limit is reached, new log files are created. ?Specify a negative number to overwrite log file when limit is reached.?
FLUSH_SIZE	Buffer size (in bytes) that EDM logger uses to accumulate log messages before writing to the EDM User log file.
MESSAGE_WIDTH	The maximum width in characters of the messages in the EDM Log.
PIPE_SIZE	Maximum memory size (in bytes) of log messages processed before logged.

Example

```
USERLOG SECTION                                                 *

*  ACTIVATE       = YES will activate this logging.  NO will not.   *

*  DIRECTORY      = Directory used to write the EDM User Log.       *

*  FLUSH_SIZE     = Buffer size in bytes that EDM:logger uses to    *

*                   accumulate log messages before writing to the   *

*                   EDM:Manager user log file. A bigger buffer size *

*                   will increase performance, but cause a delay    *

*                   in message flushing to the log file.            *

*  THRESHHOLD     = Threshold number of log messages that will be   *

*                   written to a log before automatically switching *

*                   to the next log, if it is a positive number.    *

*                   If set to a negative number the log file will   *

*                   wrap around when the limit is reached.          *

*  MESSAGE_WIDTH  = The width of the messages in the EDM User Log.  *

*                   located in the LOG directory called:            *

*                   [MGR_NAME]us[MGR_ID].log                       *


*  PIPE_SIZE      = The number of bytes to be written to a pipe     *

*                   until the messages are written to the log file. *
```

```
*-------------------------------------------------* New Example to come

[MGR_USERLOG]

ACTIVATE   = NO

DIRECTORY=e:\EDMMGR\LOG

THRESHHOLD = 5000000

FLUSH_SIZE = 256

MESSAGE_WIDTH = 256

PIPE_SIZE  = 1000000
```

Setting Name	Value as Installed	Default Value	Minimum Value
ACTIVATE	NO	NO	N/A
DIRECTORY	EDMMGR\LOG	Current directory	N/A
THRESHHOLD	-5000000 Bytes	5000 Bytes	1?
FLUSH_SIZE	256 Bytes	100000 Bytes	1?
MESSAGE_WIDTH	256 Characters	90	80?
PIPE_SIZE	1000000 Bytes	65535 Bytes	1?

Performance Considerations

The following are the performance considerations associated with the MGR_LOG section:

- Increasing the buffer size (FLUSH_SIZE) will enhance performance, but will delay messages flushed to the log file.
- Increase MESSAGE_WIDTH if log messages are being truncated.
- When modifying parameters in this section as they relate to memory or disk utilization, care must be taken to be certain the maximum amount of memory or storage space available.

4 Using EDM's Database Maintenance Programs for Windows NT

This chapter shows you how to work with the database utility programs that EDM provides to maintain the EDM Manager for Windows NT.

The EDM Manager Database Utility Programs

The EDM Manager database utilities, located in the BIN subdirectory of the EDMMGR directory, are initially used by the installation program to install the EDM Manager's database. You can then use these utilities to take a test environment database to the production environment, overlay an existing database, or move a database configuration from one operating system to another.

The following table lists and describes the database utilities.

The EDM Manager Database Utility List

Program Name	Description
EDMMDBSP	Reads a comma-separated variable (CSV) file containing multiple records, and creates an NT-usable file in binary format for each record.
EDMMRSSP	Reads a comma-separated variable file containing multiple resources, and creates an NT-usable file in binary format for each resources.
EDMMDBJO	Reverses the EDMMDBSP process, and converts all record files into one, comma-separated, variable file.
EDMMEXPOI	Exports EDM instance data (objects) from the EDM database to a third party tool for reporting purposes or for input into another database., and generates a modified comma-separated variable list of an EDM object.
EDMMEXPR	Exports EDM resources from the EDM Manager resource database, and generates a binary file that can be imported into another EDM MANAGER database (using EDMMIMPR).
EDMMEXPC	Exports an EDM class file from the EDM Manager database, and generates a binary file that can be imported into another EDM MANAGER database (using EDMMIMPC). Contents include the class template and base Instance.
EDMMIMPOI	Reads a comma-separated variable list and creates an EDM object from the list.
EDMMIMPR	Imports EDM resources into the EDM Manager database. Instances with the same names will be replaced with the input file.

Program NameY2KFIXDBEDMMIMPR	DescriptionReads and converts applicable dates within the database and control areas to ensure Year 2000 compliance.Imports EDM resources into the EDM Manager database. Instances with the same names will be replaced with the input file.
EDMMIMPC	Imports EDM class file name. Instances are updated with a time and date stamp.
Y2KFIXDB	Reads and converts applicable dates within the database and control areas to ensure Year 2000 compliance.

Warning: It is critical imperative that you periodically back up your production database when working with mission critical production applications.

Furthermore, to avoid changes to the database while you are working with its contents, do not invoke these utilities while there are users active on the system.

EDMMDBSP

EDMMDBSP is an EDM Manager method that splits and organizes files containing multiple MVS records in comma-separated variable form into directories and files that can be used by the Windows NT file system (NTFS) in binary format. The method converts data from EBSCDIC to ASCII format.

The contents of these files equate to EDM objects and instances. Each resulting file from the original configuration file comprises a single instance.

Syntax

```
EDMMDBSP database_file [record_size] [directory]
```

The default `[record_size]` is 4000.

The `[directory]` should be the directory where the EDM Manager's pPrimary file is located. To specify a different directory location, make sure that you have write access to that directory.

Specifying the EDMMDBSP Utility

The following examples show some of the ways you can specify the EDMMDBSP command options.

Examples

Command	Explanation
EDMMDBSP FBCONF.V30 4000 c:\edmdb	Separates the database file FBCONF.V30 and stores its output in the EDMDBedmdb directory.
EDMMDBSP FBCONF.V30	Separates the database file FBCONF.V30 and stores its output in the database directory defined in the EDMPROF.DAT file.

The following table describes each of the command parameters for specifying EDMM running on the Windows NT EDM Manager.

EDMMDPPSP Command Parameters

Parameter	What To Specify	Required or Optional
database_file	The database_file you are converting. In most cases, the name of the file is FBCONF.XXX. The extension of this file varies.	Required
[record_size]	This size of the file you are creating from the conversion. If you do not specify a size, the value will be extracted from the EDMPROF.DAT file. The default is 4000, or 4Kb.	Optional
[directory]	The name of the directory in which you are storing the new file, typically, the directory of your EDM database. The directory specified in the EDMPROF.DAT file will be used if none is defined.	Optional

Warning: You should run the Year 2000 fixed database utility after importing data into a database that is already Year 2000 compliant.

EDMMRSSP

EDMMRSSP is an EDM Manager method that is used to split and organize a file containing multiple resources, in comma-separated variable form, into separate resource files that can be used by the Windows NT file system in binary format.

A resource contains one or more records. The multiple records for a resource are joined to create a single file on the Windows NT file system for each resource.

Syntax

```
EDMMRSSP database_file [record_size] [directory]
```

The default `[record_size]` is 3000.

The `[directory]` should be the directory where the EDM Manager's resource file is located. To specify a different directory location, make sure that you have write access to that directory.

Specifying the EDMMRSSP Utility

The following examples show some of the ways you can specify the EDMMRSSP command options.

Examples

Command	Explanation
EDMMRSSP FBRESO.V30 3000 c:\edmdb	Separates the database file FBRESO.V30 and stores its output in the edmdb directory.
EDMMRSSP FBRESO.V30	Separates the database file FBRESO.V30 and stores its output in the database directory defined in the EDMPROF.DAT file.

The following table describes each of the command parameters for specifying EDMMRSSP running on the Windows NT EDM Manager.

EDMMRSSP Command Parameters

Parameter	What To Specify	Required or Optional
database_file	The database_file you are converting. In most cases, the name of the file is FBRESO.XXX. The extension of this file varies.	Required
[record_size]	This size of the file you are creating from the conversion. If you do not specify a size, the value will be extracted from the EDMPROF.DAT file. The default is 3000, or 3Kb.	Optional
[directory]	The name of the directory in which you are storing the new file, typically, the directory of your EDM database. The directory specified in the EDMPROF.DAT file will be used if none is defined.	Optional

Warning: You should run the Year 2000 fixed database utility after importing data into a database that is already Year 2000 compliant.

EDMMDBJO

EDMMDBJO is an EDM Manager method that reverses the EDMMDBSP process. This utility groups the objects and instances based on the domains and creates a single configuration file in comma-separated variable form, which can be used by EDM Managers on similar or different platforms. Data is converted from ASCII format to EBCDIC format.

Syntax

```
EDMMDBJO [/p] /f = EDM_file /d = EDM_domain [/c = EDM_class]

[/i = EDM_instance] /o =output_file [/r = record_size]
```

The default `[record_size]` is 4000.

You must specify an `[output_file]`. If you do not specify the `[record_size]`, the program prints the maximum record size needed to pack the database into a single file. These data can be subsequently used as the configuration file.

The `[/p]` option lists exported object displays on the screen. In addition to the exported objects, the `[/p]` option provides statistics on the number of records, classes, and instances, as well as the recommended record size and the database size (in bytes).

> **Note:** You can use prefixes and wildcards (*) for class and instance names.

Specifying the EDMMDBJO Utility

The following examples show some of the ways you can specify the EDMMDBJO command options.

Examples

Command	Explanation
EDMMDBJO **/f=** PRIMARY **/o=** FBCONG.V30	Exports all objects under the PRIMARY file to the file FBCONF.V30.
EDMMDBJO **/f=** PRIMARY **/d=** SYSTEMX **/c=** USER **/l=** DIFF* **/o=** FBCONG.V30	Exports all the objects that begin with a prefix of DIFF under the User class, Systemx domain, Primary file.
EDMMDBJO **/f=** PRIMARY **/d=** SYSTEMX **/c=** * **/o=** FBCONG.V30	Exports all the objects under the Systemx domain, Primary file.

The following table describes the command parameters for specifying EDMMDBJO running on the Windows NT EDM Manager.

EDMMDBJO Command Parameters

Parameter	What To Specify	Required or Optional
/p= EDM_	This parameter will provide you with a preview of the converted file.	Optional
/f= EDM_file	After specifying the **/f=** parameter followed by a space, specify the name of the source EDM Manager database by name. Primary, Profile, or History	Required

Parameter	What To Specify	Required or Optional
	are the only valid options. Does not allow wild cards.	
/d= EDM_domain	After specifying the **/d=** parameter followed by a space, specify the name of the source domain, such as Systemx under the Primary file. Allows wild cards; asterisk (*) only.	Required
/c= EDM_class	After specifying the **/c=** parameter followed by a space, specify the name of the source class, such as User, Dept, Workgrp or others under the Systemx domain. Allows wild cards; asterisk (*) only.	Optional
/i = EDM_instance	After specifying the **/i=** parameter followed by a space, specify the name of the source instance, such as ZCONFIG under the User class. Allows prefixes and wild cards; asterisk (*) only.	Optional
/o = output_file	After specifying the **/o=** parameter followed by a space, specify the name of the destination output file (with extension) where you want the exported data to reside. The default is the current window (sdtout).	Required
[**/r**]=	Record size.specification	Optional

Warning: You should run the Year 2000 fixed database utility after importing data into a database that is already Year 2000 compliant.

EDMMEXPOI

EDMMEXPOI is an EDM Manager method that is used to export EDM objects from the EDM database. This program generates a modified comma-separated variable (CSV) file of an EDM object. The EDMMEXPI database tool is an extension of the functionality of the original EMMEXPO tool.

EDMMEXPI - (Export Database Instance Data)

This allows you to restrict the EDM attribute data retrieved by the EDMMEXPI tool or it may retrieve all attribute data contained within the instances that you select at the command line.

The resulting modified CSV file can be imported into another EDM Manager database using the import facility EDMMIMPIO, or used to generate reports by using third party vendor software (for example, Microsoft Access, Microsoft Excel).

Note: Do not modify the exported CSV file prior to importing it into the EDM system.

To format the modified CSV output file for import into third party vendor software, you must run an EDM-provided REXX program, called EXPORT.REX, against the output file of EDMMEXPOI. This REXX will reformat the output of EDMMEXPOI into a true CSV file.

Note: EDMMEXPOI exports the EDM objects (instances) only. It does not export the classes or any underlying resources for the objects. To export the resources, refer to the EDMMEXPR program.

In generating a CSV output file, EDMMEXPOI will create unique names for connects and methods in a class. Connects can be identified by the CONNnnnn format, and methods by METHnnnn.

This is only when the "Report=Yes" switch is selected. Select "Report=No" as an option, if you intend to import to another database.

You will need to specify certain information to EDMMEXPOI and its various command line parameters when EDM is running. The literal parameters, such as **/f=**, must be specified in lower case, and the variable parameters, such as PRIMARY, must be specified in upper case.

Be sure that the Profile file, EDMPROF.DAT, is in the same directory this program is executed from. The database path is stored in the Profile file.

Syntax TO BE CHANGED TO APPROPRIATE

```
EDMMEXPOI /f= EDM_file /d= EDM_domain /c= EDM_class /i= EDM_instance /o= output_file [/p]
[/b]
```

Specifying the EDMMEXPOI Utility

The following examples show some of the possible ways you can specify the EDMMEXPOI command options.

Examples TO BE CHANGED TO APPROPRIATE EXAMPLES

Command	Explanation
EDMMEXPOI **/f=** PRIMARY **/o=** report.fil	Exports all objects under the PRIMARY file.
EDMMEXPIO **/f**= PRIMARY /d= SYSTEMX /c= USER /I= DIFF* /o= report.fil	Exports all the objects that begin with a prefix of DIFF under the User class, Systemx domain, Primary file.
EDMMEXPOI **/b /f=** PRIMARY **/d=** SYSTEMX **/c=** USER **/il=** * **/o=** report.fil	Exports all the objects under the User class, Systemx domain, Primary file, inheriting the predefined values from the base instance.
EDMMEXPOI **/p /f=** PRIMARY **/d=** SYSTEMX **/c=** USER **/i=** * **/o=** report.fil	Exports all the objects under the User class, Systemx domain, Primary file, displaying the values on the screen.
EDMMEXPOI **/f=** PRIMARY **/d=** SYSTEMX **/c=** USER **/il=** * > output.fil	Exports all the objects under the User class, Systemx domain, Primary file, displaying the values on the screen.

The following table describes the command parameters for specifying EDMMEXPO running on the Windows NT EDM Manager.

EDMMEXPOI Command Parameters

Parameter	What To Specify	Required or Optional
/f= EDM_file	After specifying the **/f=** parameter followed by a space, specify the name of the source EDM Manager database by name. Primary, Profile, or History are the only valid options. Does not allow wild cards.	Required
/d= EDM_domain	After specifying the **/d=** parameter followed by a space, specify the name of the source domain, such as Systemx under the Primary file. Allows wild cards; asterisk (*) only.	Required
/c= EDM_class	After specifying the **/c=** parameter followed by a space, specify the name of the source class, such as User, Dept, Workgrp or others under the Systemx domain. Allows wild cards; asterisk (*) only.	Required
/i = EDM_instance	After specifying the **/i=** parameter followed by a space, specify the name of the source instance, such as ZCONFIG under the User class. Allows prefixes and wild cards; asterisk (*) only.	Required
/o = output_file	After specifying the **/o=** parameter followed by a space , specify the name of the destination output file (with extension) where you want the exported	Required

Parameter	What To Specify	Required or Optional
	data to reside. The default is the current window (sdtout).	
[**/b**]	Specify this parameter if you want to include all the attributes that are predefined from the base instance.	Optional
[**/p**]	Specify this parameter if you want to display the exported objects on the console.	Optional

EDMMEXPI for NT

Usage: EDMMEXPI PREVIEW=YES/NO,FILE=file_name,DOMAIN= domain_name,CLASS=class_name,FROMINST=from_inst,TOINST=to_inst,FROMDATE= from_date,FROMTIME=from_time,TODATE=to_date,TOTIME=to_time,KEEP=keep_list_ file,DROP=drop_list_file,OUTPUT=output_file,COMMENT=comment,REPORT=YES/NO, ORDER=YES/NO**Warning:** You should run the Year 2000 fixed database utility after importing data into a database that is already Year 2000 compliant.

EDMMEXPR

EDMMEXPR is an EDM Manager method that exports EDM resources from the EDM Manager resource database. The resulting file can be imported into another EDM Manager database using the EDMMIMPR import facility.

EDMMEXPR - (Export Database Resource Data)

There is specific information you will need to specify to EDMMEXPR and its various command line parameters when EDM is running. The literal parameters, such as **/f=**, must be specified in lower case, and the variable parameters, such as PRIMARY, must be specified in upper case.

The Profile file, EDMPROF.DAT, should be in the same directory this program is executed from. The database path is stored in the Profile file.

EDMMEXPR is used to join resources data from the EDM configuration file (eg PRIMARY) into a single platform independent binary file that may be utilized between EDM Manager platforms

Syntax - NOT CORRECT SYNTAX MUST BE UPDATED

EDMMEXPR /f= EDM_file **/d=** EDM_domain **/c=** EDM_class **/i=** EDM_instance **/o=** output_file [**/p**] [**/r=** record_size]

Specifying the EDMMEXPR Utility

The following examples show some of the possible ways you can specify the EDMMEXPR command options.

Examples - TABLE INACCURATE - NEEDS UPDATE TO UTILITY

Command	Explanation
EDMMEXPR **/f=** PRIMARY **/o=** MYFILE.bin	Exports all instances under the PRIMARY file.
EDMMEXPR **/p /f=** PRIMARY **/i=** "TSO*" **/o=** MYFILE.bin	Exports all the instances that begin with a prefix of TSO under the Primary file.

Command	Explanation
EDMMEXPR **/f=** PRIMARY **/d=** SYSTEMX **/c=** ZRSOURCE **/o=** MYFILE.bin **/r=** 5000	Exports all the instances under the Primary file, SYSTEMX domain, ZRSOURCE class using a record size of 5000.
EDMMEXPR **/p /f=** PRIMARY **/o=** MYFILE.bin	Creates a screen preview of the expected export.
EDMMEXPR **/p /f=** PRIMARY **/o=** MYFILE.bin > preview.fil**EDMMEXPR /f=** PRIMARY **/d=** SYSTEMX **/c=** ZRSOURCE **/o=** MYFILE.bin **/r=** 5000	Creates a preview of the expected export into a file.Exports all the instances under the Primary file, SYSTEMX domain, ZRSOURCE class using a record size of 5000.

The following table describes the command parameters for specifying EDMMEXPR running on the Windows NT EDM Manager.

EDMMEXPR Command Parameters

TABLE INACCURATE - NEEDS CORRECT INFO

Parameter	What To Specify	Required or Optional
/f= EDM_file	After specifying the **/f=** parameter followed by a space, specify the name of the source EDM Manager database by name. Primary is the only valid option. Does not allow wild cards.	Required
/d= EDM_domain	After specifying the **/d=** parameter followed by a space, specify the name of the source domain, such as Systemx under the Primary file. Allows wild cards; asterisk (*) only.	Required
/c= EDM_class	After specifying the **/c=** parameter followed by a space, specify the name of the source class, such as User, Dept, Workgrp or others under the Systemx domain. Allows wild cards; asterisk (*) only.	Required
/i = EDM_instance	After specifying the **/i=** parameter followed by a space, specify the name of the source instance, such as ZCONFIG under the User class. Allows prefixes and wild cards; asterisk (*) only.	Required
/o = output_file	After specifying the **/o=** parameter followed by a space, specify the name of the destination output file (with extension) where you want the the exported data to reside. The default is the current window (sdtout).	Required
[**-p**]	Specify this parameter if you want a preview listing of what resources would be exported. This can be piped to an output file for printing.	Optional
[**-r** xxxx][**-p**]	Specify this parameter followed by a number to indicate the record size to be used when joining the resources. The default value is 3000. This parameter is used by the corresponding program EDMMIMPR to break up the file into its constituent pieces. Note that you must use the same number for both the import and export of resources from the EDM configuration files.Specify this parameter if you want a preview listing of what resources would be exported. This can be piped to an output file for printing.	OptionalOptional

Warning: You should run the Year 2000 fixed database utility after importing data into a database that is already Year 2000 compliant.

EDMMEXPC

EDMMEXPCR is an EDM Manager method that exports EDM classes from the EDM Manager database. The resulting file contains the exported class templates and base instances and can be imported into another EDM Manager database using the EDMMIMPC import facility.

There is specific information you will need to specify to EDMMEXPCR and its various command line parameters when EDM is running. The literal parameters, such as **/f=**, must be specified in lower case, and the variable parameters, such as PRIMARY, must be specified in upper case.

The Profile file, EDMPROF.DAT, should be in the same directory this program is executed from. The database path is stored in the Profile file.

Syntax

```
EDMMEXPCR /f= EDM_file /d= EDM_domain /c= EDM_class [/p= preview]

/o= output_file [/c= comment]
```

Specifying the EDMMEXPC Utility

The following examples show some of the possible ways you can specify the EDMMEXPC command options.

Examples

Command	Explanation
EDMMEXPC **/f=** PRIMARY **/o=** MYFILE.bin	Exports all instances under the PRIMARY file.
EDMMEXPC **/f=** PRIMARY **/d=** SYSTEMX **/c=** ZRSOURCE **/o=** MYFILE.bin	Exports all the instances under the Primary file, SYSTEMX domain, ZRSOURCE class..
EDMMEXPC **/p /f=** PRIMARY **/o=** MYFILE.bin	Creates a screen preview of the expected export.
EDMMEXPC **/p /f=** PRIMARY **/o=** MYFILE.bin > preview.fil	Creates a preview of the expected export into a file.
EDMMEXPC **/p /f=** PRIMARY **/c=** My comment here	Inserts a comment together with the output in a file

The following table describes the command parameters for specifying EDMMEXPC running on the Windows NT EDM Manager.

EDMMEXPCR Command Parameters

Parameter	What To Specify	Required or Optional
/f= EDM_file	After specifying the **/f=** parameter followed by a space, specify the name of the source EDM Manager database by name. Primary is the only valid option. Does not allow wild cards.	Required
/d= EDM_domain	After specifying the **/d=** parameter followed by a space, specify the name of the source domain, such as Systemx under the Primary file. Allows wild cards; asterisk (*) only.	Required
/c= EDM_class	After specifying the **/c=** parameter followed by a space, specify the name of the source class, such as User, Dept, Workgrp or others under the Systemx domain. Allows wild cards; asterisk (*) only.	Required
/o = output_file	After specifying the **/o=** parameter followed by a space, specify the name of the destination output file (with extension) where you want the the exported data to reside. The default is the current window (sdtout).	Required
[**-p**]	Specify this parameter if you want a preview listing of what resources would be exported. This can be piped to an output file for printing.	Optional
[**-c**]	Specify this parameter I you want to add a comment .	Optional

Warning: You should run the Year 2000 fixed database utility after importing data into a database that is already Year 2000 compliant.

EDMMIMPOI

EDMMIMPO EDMMIMPI is an EDM Manager method that is used to read a comma-separated variable list and create an EDM object from the list.

This section shows you how to specify EDMMIMPOI and its various command line parameters. The literal parameters, such as **/f=**, must be specified in lower case, and the variable parameters, such as MYFILE, must be specified in upper case.

Be sure the Profile file, EDMPROF.DAT, is in the same directory this program is executed from. The database path is stored in the Profile file.

Syntax

EDMMIMPOI /f= input_file [**/p**] [**/d**]

Specifying the EDMMIMPOI Utility

The following examples show some of the ways you can specify the EDMMIMPOI command options.

Examples

Command	Explanation
EDMMIMPOI **/f =** MYFILE	Imports a file into an EDM Manager.
EDMMIMP**OI /p /f**= myfile	Creates a preview listing of input file contents.
EDMMIMPOI **/p /f** = MYFILE > preview.fil	Creates a preview listing of the input file contents piped into an output file for printing.
EDMMIMPOI **/d /f =** MYFILE	Creates a debug listing of all the variables and their values in the import file.
EDMMIMPI**O /d /f=** MYFILE > debug.fil	Creates a debug listing of all the variables and their values in the import file piped into an output file for printing.

Note: Variable names that are present in the input data file but not defined in the class template are dropped when the EDM object is created.

The following table describes the command parameters for specifying EDMMIMPIO running on the Windows NT EDM Manager.

EDMMIMPOI Command Parameters

Parameter	What To Specify	Required or Optional
/f= input_file	After specifying the **/f=** parameter followed by a space, specify the file containing the input information; that is, the EDM object you want to create or update.	Required
[**/p**]	Specify this parameter if you want a preview listing of what the input file contains. This can be piped to an output file for printing.	Optional
[**/d**]	Specify this parameter if you want a preview listing of the variables and their values in the input file. This can be piped to an output file for printing.	Optional
[**/t**]	Allows you to change the date, time or object id.	Optional
[**/t**] [**/d**]	To replace existing instances.Specify this parameter if you want a preview listing of the variables and their values in the input file. This can be piped to an output file for printing.	Optional

Warning: You should run the Year 2000 fixed database utility after importing data into a database that is already Year 2000 compliant.

EDMMIMPR

EDMMIMPR is an EDM Manager method that is used to import resources into an EDM Manager database. All existing resources with the same name will be replaced by the input file.

This section shows you how to specify EDMMIMPR and its various command line parameters when EDM is running under Windows NT. The literal parameters, such as **/f=**, must be specified in lower case, and the variable parameters, such as MYFILE, must be specified in upper case.

The Profile file, EDMPROF.DAT, should be in the same directory this program is executed from. The database path is stored in the Profile file.

Syntax

```
EDMMIMPR /f= input_file [/p] [/d] [-r xxxx]
```

Specifying the EDMMIMPR Utility

The following examples show some of the ways you can specify the EDMMIMPR command options.

Examples

Command	Explanation
EDMMIMPR /f MYFILE.bin	Imports a file into an EDM Manager.
`EDMMIMPR /f myfile.bin /r= 5000`	Creates a preview listing of input file contents.
`EDMMIMPR /f myfile.bin /r= 3000 hp/dbase/resource`	Creates a preview listing of the input file contents piped into an output file for printing.

The following table describes the command parameters for specifying EDMMIMPR running on the Windows NT EDM Manager.

EDMMIMPR Command Parameters

Parameter	What To Specify	Required or Optional
/f input_file	After specifying the **/f** parameter followed by a space, specify the file containing the input information; that is, the EDM object you want to create or update.	Required
[**/p**]	Specify this parameter if you want a preview listing of the resources in the input file. This can be piped to an output file for later printing.	Optional
[**/d**]	Specify this parameter if you want a preview listing of the variables and their values in the input file. This can be piped to an output file for printing.	Optional
[**-r** xxxx]	Specify this parameter followed by a number to indicate the record size to be used when joining the resources. The default value is 3000. This parameter is used by the corresponding program EDMMIMPO. Note that you must use the same number for both the import and export of resources from the EDM configuration files.	Optional

Warning: You should run the Year 2000 fixed database utility after importing data into a database that is already Year 2000 compliant.

EDMMIMPC

EDMMIMPC is an EDM Manager method that is used to import class files into an EDM Manager database. All existing resources with the same name will be replaced by the input file. Is this true?

This section shows you how to specify EDMMIMPC and its various command line parameters when EDM is running under Windows NT. The literal parameters, such as **/f=**, must be specified in lower case, and the variable parameters, such as MYFILE, must be specified in upper case.

The Profile file, EDMPROF.DAT, should be in the same directory this program is executed from. The database path is stored in the Profile file.

Syntax

```
EDMMIMPC /f= input_file [/p] [/d] [-r xxxx]
```

Specifying the EDMMIMPC Utility

The following examples show some of the ways you can specify the EDMMIMPC command options.

Examples

Command	Explanation
EDMMIMPC /f MYFILE.bin	Imports a file into an EDM Manager.
`EDMMIMPC /f myfile.bin /r= 3000`	Creates a preview listing of input file contents.
`EDMMIMPC /f myfile.bin /r= 3000 hp/dbase/resource`	Creates a preview listing of the input file contents piped into an output file for printing.

The following table describes the command parameters for specifying EDMMIMPC running on the Windows NT EDM Manager.

EDMMIMPC Command Parameters

Parameter	What To Specify
PREVIEW	• **YES** -Previews the expected results in the edmmexpi.log. No import will be performed. The default is YES. • **NO** - Applies the database changes contained in the input_file_name.
FILE	The keyword file should be followed by the filename whichthat contains the output from the execution of the EDMMEXPC command.
TIME	• **OLD** - Retains class object time stamps and Object ID. The default is OLD. • **NEW** - Updates the class time stamp and the object id based on the import.
REPLACE	• **YES** - Replaces the class template if it already exists. The old class template and ALL instances will be deleted. The default is YES. • **NO** - If the class already exists, it will not be modified
XDF	This flag only has significance when you are creating a new database. • **YES** - Imports the class template in expanded form. The default is YES. • **NO** - Imports the class in a the old Novadigm non-expanded database format if the database already exists. The expanded database format became effective with NT Manager Build 21 (Version 3.2) and higher. If you are using an NT Manager Version 4.0 you should import with the

Parameter	What To Specify
	XDF=YES option. The format of the imported class will be decided by the format of the existing database.

Warning: Caution should be used with the REPLACE = YES option. If Yyou choose this option, all existing instances in the file being imported to will be deleted.should run the Year 2000 fixed database utility after importing data into a database that is already Year 2000 compliant.

Warning: You should run the Year 2000 fixed database utility after importing data into a database that is already Year 2000 compliant.

Y2KFIXDB

Y2KFIXDB is an EDM database utility that modifies areas in the EDM database to make the entire database Year 2000 compliant. This utility searches out and converts only the appropriate dates within the database, such as ZOBJDATE, ZRSCDATE, and dates in the control areas. After this utility is run, the database may be considered Year 2000 compliant.

Y2KFIXDB is run from the command line, from the current directory if it has visibility to the EDMPROF.DAT file. There are no parameters to specify with this utility. This utility can be run as often as required.

The results of running the Y2KFIXDB utility are reflected in the EDM Manager log and in the Y2KFIXDB.LOG, located in the current directory.

Note: If you are using one of the database import facilities described in this chapter, you should run Y2KFIXDB after the data is imported.

During EDM Client processing, the EDM Manager will determine if individual EDM clients are Year 2000 compliant. The EDM Manager will proceed with Client Resolution if the EDM Client is not Year 2000 compliant. However, objects used in any database operation will be made Year 2000 compliant, but objects returned to the EDM Client will be in the format that the Client can use (i.e., non Year 2000 compliant.)

Note: The Version 4.05 EDM Manager will have a fully compliant Year 2000 database.

5 Using the EDM Manager Logging Facility

This chapter explains the naming conventions and formats used in the activity logs generated by the EDM Manager for the Windows NT logging facility, and how trace settings in the Windows NT file affect logging activity. Also, this chapter explains how to interpret and understand the EDM Manager log and the messages it contains. See Chapter 6,: " *EDM Manager Messages,*", to look up specific messages and their meanings.

An Overview of the EDM Logging Facility

The EDM MANAGER Manager for Windows NT maintains an activity log that you can use for informational purposes and for problem determination. These logs are generated for EDM Manager activity.

Please note that the EDM Manager log is different from the NT Event Log, both in format and content. The NT Event Log provides information on EDM Manager startup and shutdown only, while the EDM Manager log reports a wide range of EDM program activity. This includes everything from Manager startup to the logging on of each EDM Client, and the establishment of EDM Class Browser sessions. As with the EDM Client logs, you can customize the tracing levels of diagnostic logging for the EDM Manager.

The volume and detail level of the messages that your EDM Manager produces depends on two factors, trace settings and the logging options specified for your EDM Manager.

With EDM Release 3.0x, all EDM Manager messages are marked by an alphanumeric character string that begins with the prefix “EDM”, followed by the message number (xxxx), and ends with a one-character identifier. The one-character suffix—(I) informational, (E) error, or (W) warning—indicates the type of message being logged. This should help you to effectively analyze the log activity.

Specifying the EDM Manager Log Location and Settings

During the installation process, you allocate space for the EDM Manager log. After the installation process, you must specify the location of the log file in the MGR_LOG section of the EDMPROF.DAT file.

The MGR_LOG section of the Profile file contains five settings that determine where the log is located and how ‘elastic’ it is:

- The Directory setting tells you the drive and directory to which the log will be written.
- The FLUSH_SIZE setting determines the buffer size (in bytes) used by the EDM Manager to store messages before actually writing them to the log.
- The THRESHOLD setting displays how many lines the log file will contain before it is dumped to another file. When the log is dumped, it will be given a temporary file name in the format of EDMMRxxx. A negative threshold value will reuse the same file, overwriting the previous log segment.
- The PIPESIZE setting determines how many messages (in bytes) the EDM Manager collects before processing and storing them in the log buffers.
- The MESSAGE_WIDTH setting displays the length (in characters) of each message line.

As with most EDM parameters, the EDM Manager log settings can be tuned for maximum efficiency after monitoring your system’s performance.

The log will accumulate messages to fill up the allocated space. When this threshold is reached, the EDM Manager will create a new log, designated by a different log ID, or write over the previous log. If this occurs, you can change the Trace settings to capture fewer messages, or increase the allocated size of the log. Also, you can change the MGR_LOG settings, if warranted, to maximize memory usage and for other performance considerations. When a new log file is created, it is identified by an EDMMR prefix, and a three digit extension.

For details about specifying the MGR_LOG settings, see Chapter 3: 3*, "Tuning the EDM Manager for Windows NT*."

Controlling Trace Settings

The number of trace settings you specify will influence the number of log messages that are written to the EDM Manager. l

The TRACE section of the EDMPROF.DAT file controls and influences the diagnostic logging for the EDM Manager. This section contains keywords that represent trace settings. All diagnostic output produced by TRACE keywords is written to the active EDM Manager log. To specify a trace setting, specify YESYes. To deselect a trace setting, specify NONo.

The TRACE keywords you specify in the trace section are invoked at EDM Manager initialization and remain in effect until you change them by modifying the TRACE section, and then restarting the EDM Manager. The trace settings in effect at EDM Manager initialization are displayed at the beginning of the EDM Manager log.

For details about specifying the TRACE keywords, see Chapter 3, "*Tuning the EDM Manager for Windows NT*."

Chapter 3: Tuning the EDM Manager for Windows NT.

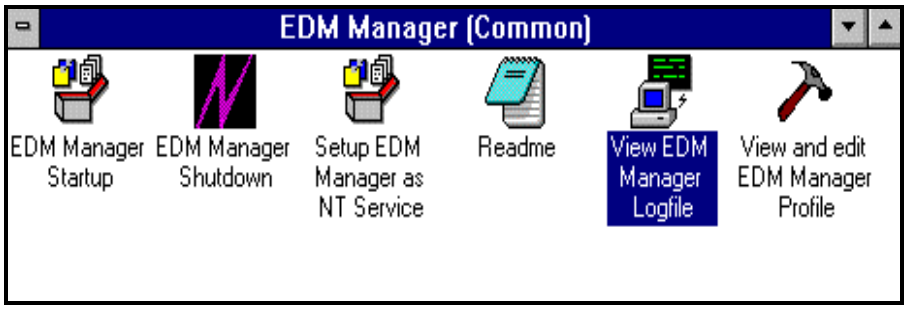
Viewing the EDM Manager Log

The EDM Manager log file reports on any EDM program activity. These messages can be viewed by following the instructions below.

Note: the EDM Manager log file is located in the LOG subdirectory of the EDMMGR directory.

➢ **To View the EDM Manager Logfile (Windows NT 3.51):**

1. Choose the EDM Manager program group located in the Program Manager.
2. The **EDM Manager (Common)** program group box is displayed:.



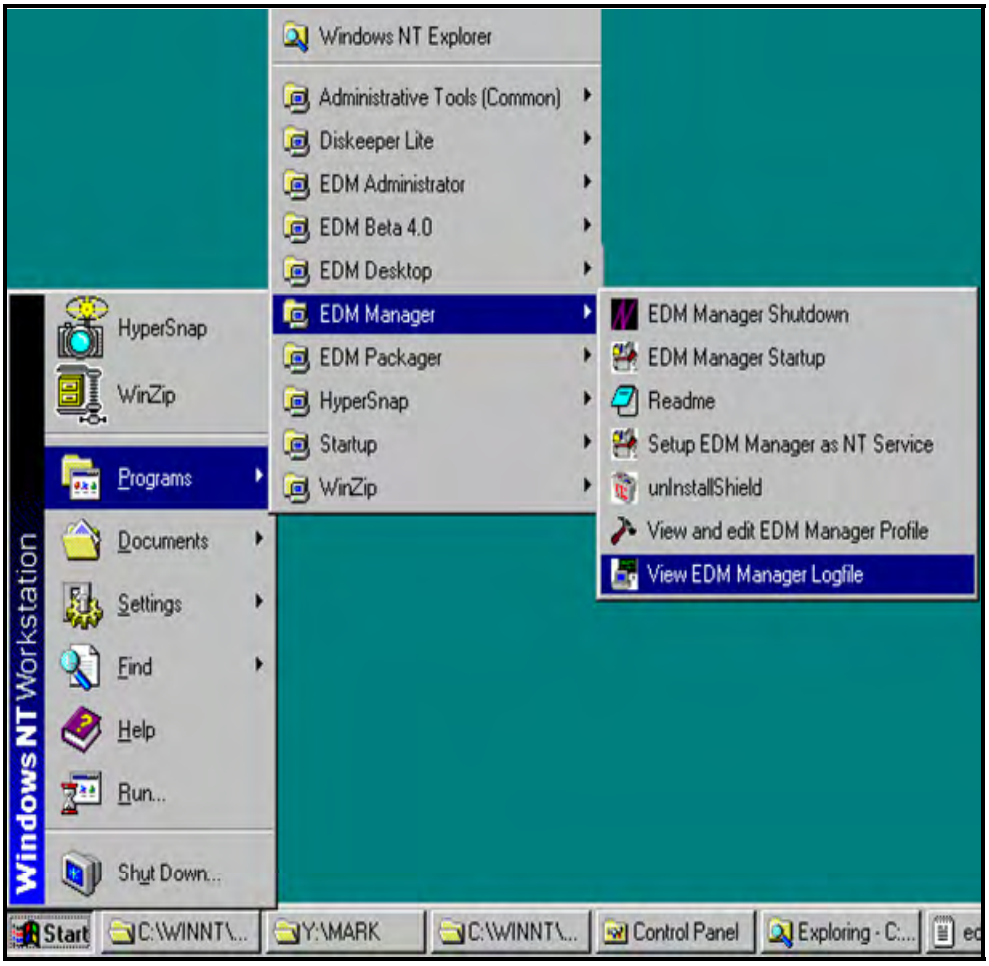
3. Choose the View EDM Manager Logfile icon.

➢ **To View the EDM Manager Logfile (Windows NT 4.0):**

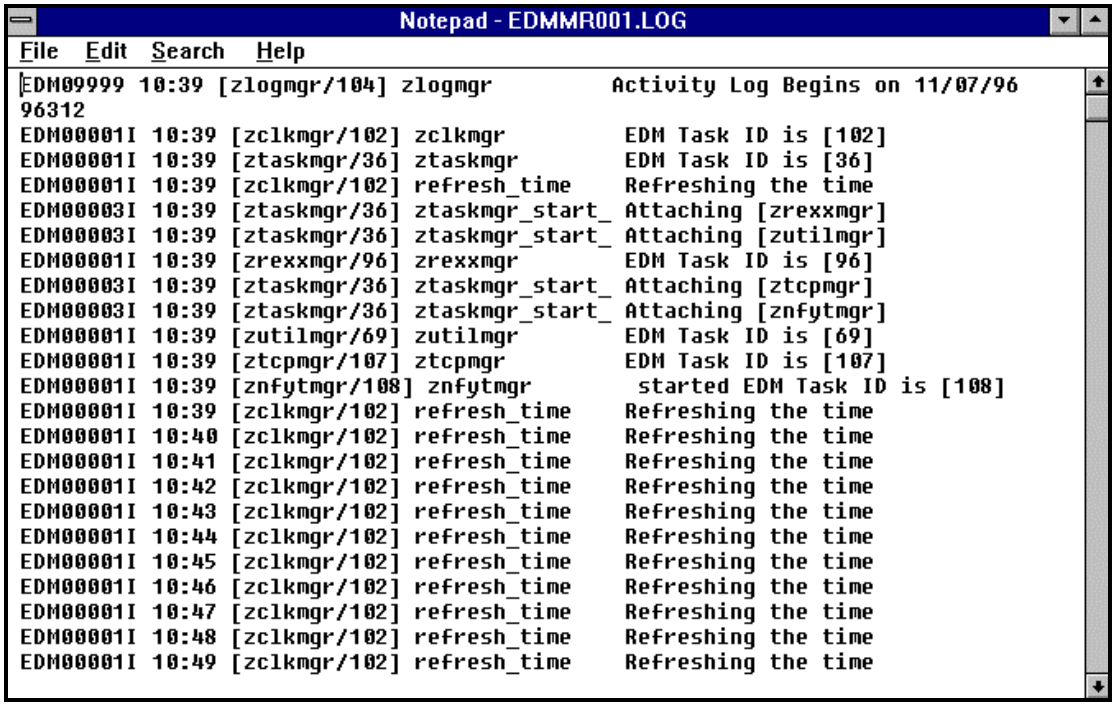
1 Select the Programs folder from your Start menu.

2 Then, select EDM Manager from the next program list.

3 Choose View EDM Manager Logfile.



The initial EDM Manager Logfile (EDMMR001.LOG) window is displayed:



This window displays the EDM Manager Logfile as recorded by the EDM Manager. The activity log contains messages pertaining to activities as specified in your TRACE section of the EDMPROF.DAT file.

After you have reviewed the log file, close the EDM Manager Logfile panel by choosing Exit from the File menu.

For additional information about the EDM Manager Logfile, see *"Reading the EDM Manager Log"* and *"An EDM An EDM Manager Sample Activity Log."* below.

Reading the EDM Manager Log

The EDM Manager logging facility generates an activity log with multiple messages (or line entries). To help you read these entries and understand the information, you should be familiar with the format used. A typical line entry is displayed in the following sample from the EDM Manager for Windows NT log.

Example

```
EDM00001I 11:51 zutilmgr UTILITY MANAGER TASK HAS STARTED
```

This message consists of a single line entry with four log fields, as follows:

- The **Message Number** (EDMnnnn) denotes the EDM Message number. Note that the message number ends with a letter to mark the message as an information (I), error (E), or warning (W) message.
- The Time Stamp (hh:mm) indicates the time that the event occurred.
- The Taskid (xxxxxxxx) indicates the task that the activity is being logged for. This can be an EDM system task, such as zutilmgr.
- The **Message** (UPPER CASE LETTERS) is the actual message text.

Note: Typically, activity logs contain multiple messages that represent an activity. Most activities that require multiple messages will be framed with an **activity begins** and an **activity ends** message.

An EDM Manager Sample Activity Log

A properly sized and configured log will give you appropriate detail on the entire scope of EDM Manager operations. While individual events vary from installation to installation, the general flow of the Manager log consists of four phases:

- Manager startup
- Client Connect
- Object resolution
- Manager shutdown

The following sections describe the types of messages that result from each phase.

Manager Startup

Manager startup is reflected in a series of messages at the beginning of the log, starting with the actual opening of the log itself. Next, the EDM Manager will process the parameters in the EDMPROF.DAT file and start various Manager tasks. At this point, the EDM Manager is fully configured and operational, waiting for EDM Clients to connect to continue processing. Messages associated with this phase include Manager tasks starting, Manager tasks attaching, and Profile section names and values.

Client Connect

The EDM Client Connect process occurs when an EDM Client requests a session with the EDM Manager to be resolved to its desired state. Each Client Connect is itself a Manager task. The EDM Manager may defer the session to a time when it is not occupied, or will connect to the EDM Client at the time of the request. Messages associated with Client Connect include “Client task has started” or “Connected to client.”

Object Resolution

Object resolution follows a successful Client Connect. The EDM Manager processes EDM Client objects, performs methods on those objects, transfers data to and from the EDM Client, and creates objects containing audit and profile information for that Client. Then, when the Client has been returned to its desired state, the EDM Manager releases the Client and waits for another connect request. Messages associated with the object resolution phase include “Object resolution (object name) begins/ends”, “Current path (pointing to the methods directory or resource file)”, warnings that objects could not be found or duplicate objects exist, method identification, and return codes for methods.

Manager Shutdown

Manager shutdown occurs when the EDM Manager has serviced all EDM Clients requesting connections, has been halted, or was prematurely stopped due to an error condition. Messages associated with this phase include “Shutting down EDM Manager”, messages halting communications, or error messages that result in the EDM Manager ending operations.

Sample Activity Log

The following activity log, excerpted from an EDM Manager for Windows NT, shows a six-minute portion of program activity related to starting and stopping the EDM Manager. This example shows multiple messages that were logged as a result of processing a variety of system tasks—including:

- Opening the EDM Manager log.
- Starting the EDM Manager.
- Processing the EDMPROF.DAT settings.
- Attaching various EDM Manager programs.
- Connecting to a TCP/IP session.
- Attaching an EDM Client.
- Processing the EDM Client task.

Then, the log shows the EDM Manager for Windows NT shutting down or closing the respective tasks and databases.

For greater detail on the messages and activities logged in this example, see *Chapter 6: " EDM Manager Messages",* Chapter 6.

Example

```
EDM9999I 09:34 [zlogmgr  /118]   Activity Log Begins on 05/29/97 97149

EDM1121I 09:34 [ztoptask /065]
======================================================================

EDM1121I 09:34 [ztoptask /065]   Manager Information Section

EDM1121I 09:34 [ztoptask /065]
======================================================================

EDM0604I 09:34 [ztoptask /065]   EDM Manager is Version [4.0.1B];

                                 Build [22]

EDM1121I 09:34 [ztoptask /065]
======================================================================

EDM1121I 09:34 [ztoptask /065]   System Information Section

EDM1121I 09:34 [ztoptask /065]
======================================================================

EDM0604I 09:34 [ztoptask /065]   Windows NT Version [4.0];  Build

                                 Number [1381]; CSD []

EDM0604I 09:34 [ztoptask /065]   There is 1 Processor of type [Intel

                                 Pentium Model 2, Stepping 5]

EDM0604I 09:34 [ztoptask /065]   Physical Memory: Total=[32952320]

                                 Available=[9162752]
```

```
EDM0604I 09:34 [ztoptask /065]  Virtual  Memory: Total=[181293056]
                                  Available=[157519872] Page  size=[4096]
EDM0604I 09:34 [ztoptask /065]   NT Foreground Priority Boost is [0]
EDM0604I 09:34 [ztoptask /065]   Running on Domain [] Computer [EDM40]
                                as User [SYSTEM]
EDM0604I 09:34 [ztoptask /065]   System Directory is
                                [C:\WINNT\System32]
EDM0604I 09:34 [ztoptask /065]   Windows Directory is [C:\WINNT]
EDM0604I 09:34 [ztoptask /065]   EDM Database is on [Fixed] [NTFS]
                                [Compressed] Drive [C]
EDM0604I 09:34 [ztoptask /065]   Drive [C] supports [255] character
                                file names
EDM0604I 09:34 [ztoptask /065]   Drive [C] supports [Case-Sensitivity
                                Case-Preservation Unicode ]
EDM0604I 09:34 [ztoptask /065]   Database resides in [C:\EDMDB\]
EDM0604I 09:34 [ztoptask /065]   Database Disk Space
                               ToTotal=[1622573056]


                                Available=[640880128]
EDM0604I 09:34 [ztoptask /065]  Log will be created in in
                              C:\EDMMGR\LOG\]
EDM0604I 09:34 [ztoptask /065]  Log Disk Space Total=[1622573056]
                                Available=[640880128]

EDM1121I 09:34 [ztoptask /065]
================================================================

EDM1121I 09:34 [ztoptask /065]  Profile Section(MGR_STARTUP)

EDM1121I 09:34 [ztoptask /065]
================================================================

EDM0604I 09:34 [ztoptask /065]  Manager_TYPE         ==> DISTRIBUTED
EDM0604I 09:34 [ztoptask /065]  MGR_ID               ==> 999
EDM0604I 09:34 [ztoptask /065]  MGR_NAME             ==> EDM
EDM0604I 09:34 [ztoptask /065]  TCP_PORT             ==> 7860

EDM1121I 09:34 [ztoptask /065]
================================================================

EDM1121I 09:34 [ztoptask /065]  Profile Section(MGR_TIMEOUT)
```

```
EDM1121I 09:34 [ztoptask /065]
======================================================================

EDM0604I 09:34 [ztoptask /065]  TIMEOUT_COMM         ==> 300

EDM0604I 09:34 [ztoptask /065]  TIMEOUT_NCOMM        ==> 300

EDM0604I 09:34 [ztoptask /065]  ADMIN_TIMEOUT        ==> 0

EDM1121I 09:34 [ztoptask /065]
======================================================================

EDM1121I 09:34 [ztoptask /065]  Profile Section(MGR_TPINIT)

EDM1121I 09:34 [ztoptask /065]
======================================================================

EDM0604I 09:34 [ztoptask /065]  BUFTCP               ==> 12288

EDM0604I 09:34 [ztoptask /065]  BUFLU2               ==> 900

EDM0604I 09:34 [ztoptask /065]  BUFLU62              ==> 12288

EDM0604I 09:34 [ztoptask /065]  BUFSIPX              ==> 10240

EDM0604I 09:34 [ztoptask /065]  MAXREC               ==> 6144

EDM1121I 09:34 [ztoptask /065]
======================================================================

EDM1121I 09:34 [ztoptask /065]  Profile Section(MGR_TRACE)

EDM1121I 09:34 [ztoptask /065]
======================================================================

EDM0604I 09:34 [ztoptask /065]  ADMIN                ==> NO

EDM0604I 09:34 [ztoptask /065]  ADMPROM              ==> NO

EDM0604I 09:34 [ztoptask /065]  ALL                  ==> NO

EDM0604I 09:34 [ztoptask /065]  ALLOC                ==> NO

EDM0604I 09:34 [ztoptask /065]  AUDIT                ==> NO

EDM0604I 09:34 [ztoptask /065]  BUFF                 ==> NO

EDM0604I 09:34 [ztoptask /065]  CMPR                 ==> NO

EDM0604I 09:34 [ztoptask /065]  COMM                 ==> NO

EDM0604I 09:34 [ztoptask /065]  COMMDATA             ==> NO

EDM0604I 09:34 [ztoptask /065]  COMMCBS              ==> NO

EDM0604I 09:34 [ztoptask /065]  CONFIG               ==> NO

EDM0604I 09:34 [ztoptask /065]  CPIC                 ==> NO

EDM0604I 09:34 [ztoptask /065]  DATA                 ==> NO

EDM0604I 09:34 [ztoptask /065]  DSCOMP               ==> NO
```

```
EDM0604I 09:34 [ztoptask /065]  ENQDEQ              ==> NO

EDM0604I 09:34 [ztoptask /065]  EXPL                ==> NO

EDM0604I 09:34 [ztoptask /065]  FILE                ==> NO

EDM0604I 09:34 [ztoptask /065]  IMPL                ==> NO

EDM0604I 09:34 [ztoptask /065]  LOOKASID            ==> NO

EDM0604I 09:34 [ztoptask /065]  METHOD              ==> NO

EDM0604I 09:34 [ztoptask /065]  NOTIFY              ==> NO

EDM0604I 09:34 [ztoptask /065]  OBJCRC              ==> NO

EDM0604I 09:34 [ztoptask /065]  OBJRES              ==> NO

EDM0604I 09:34 [ztoptask /065]  OBJRES0             ==> NO

EDM0604I 09:34 [ztoptask /065]  OBJRES1             ==> NO

EDM0604I 09:34 [ztoptask /065]  OBJXFER             ==> NO

EDM0604I 09:34 [ztoptask /065]  PASSWORD            ==> NO

EDM0604I 09:34 [ztoptask /065]  PROFILE             ==> NO

EDM0604I 09:34 [ztoptask /065]  PROMOTE             ==> NO

EDM0604I 09:34 [ztoptask /065]  RESOURCE            ==> NO

EDM0604I 09:34 [ztoptask /065]  REXX                ==> NO

EDM0604I 09:34 [ztoptask /065]  REXXOFF             ==> NO

EDM0604I 09:34 [ztoptask /065]  STATS               ==> NO

EDM0604I 09:34 [ztoptask /065]  SUBST               ==> NO

EDM0604I 09:34 [ztoptask /065]  TCP                 ==> NO

EDM0604I 09:34 [ztoptask /065]  TEST                ==> NO

EDM0604I 09:34 [ztoptask /065]  TRAN                ==> NO

EDM0604I 09:34 [ztoptask /065]  VAR                 ==> NO

EDM0604I 09:34 [ztoptask /065]  VARSTG              ==> NO

EDM0604I 09:34 [ztoptask /065]  VSAM                ==> NO

EDM0604I 09:34 [ztoptask /065]  VSAMDATA            ==> NO

EDM0604I 09:34 [ztoptask /065]  VSAMRPLS            ==> NO

EDM1121I 09:34 [ztoptask /065]
================================================================

EDM1121I 09:34 [ztoptask /065]  Profile Section(MGR_RETRY)

EDM1121I 09:34 [ztoptask /065]
================================================================
```

```
EDM0604I 09:34 [ztoptask /065]  BUSY_RETRY           ==> 1

EDM0604I 09:34 [ztoptask /065]  DISA_RETRY           ==> 999

EDM1121I 09:34 [ztoptask /065]
======================================================================

EDM1121I 09:34 [ztoptask /065]  Profile Section(MGR_NOTIFY)

EDM1121I 09:34 [ztoptask /065]
======================================================================

EDM0604I 09:34 [ztoptask /065]  NFYT_TIMEOUT         ==> 60

EDM0604I 09:34 [ztoptask /065]  NFY6_TPN             ==> EDMINITN

EDM1121I 09:34 [ztoptask /065]
======================================================================

EDM1121I 09:34 [ztoptask /065]  Profile Section(MGR_LICENSE)

EDM1121I 09:34 [ztoptask /065]
======================================================================

EDM0604I 09:34 [ztoptask /065]  LICENSE_STRIN        ==> F904A1F206E18EE8 8A08A2D1A2E7D840
FBD40910A2571198 109298BCC

EDM1121I 09:34 [ztoptask /065]
======================================================================

EDM1121I 09:34 [ztoptask /065]  Profile Section(MGR_ATTACH_LIST)

EDM1121I 09:34 [ztoptask /065]   ============================================

EDM0604I 09:34 [ztoptask /065]  ATTACH_LIST_SLOTS    ==> 15

EDM0604I 09:34 [ztoptask /065]  RESTART_LIMIT        ==> 7

EDM0604I 09:34 [ztoptask /065]  VERIFY_INTERVAL      ==> 5

EDM0604I 09:34 [ztoptask /065]  CMD_LINE             ==> (zutilmgr)

                                RESTART=YES

EDM0604I 09:34 [ztoptask /065]  CMD_LINE             ==> (zrexxmgr)

                                RESTART=YES

EDM0604I 09:34 [ztoptask /065]  CMD_LINE             ==> (ztcpmgr)
                                RESTART=YES

EDM1121I 09:34 [ztoptask /065]
======================================================================

EDM1121I 09:34 [ztoptask /065]  Profile Section(MGR_ACCESS)

EDM1121I 09:34 [ztoptask /065]
======================================================================

EDM0604I 09:34 [ztoptask /065]  ADMIN                ==> deny

EDM0604I 09:34 [ztoptask /065]  CONSOLE              ==> deny
```

```
EDM1121I 09:34 [ztoptask /065]
======================================================================

EDM1121I 09:34 [ztoptask /065]  Profile Section(MGR_LOG)

EDM1121I 09:34 [ztoptask /065]
======================================================================

EDM0604I 09:34 [ztoptask /065]  DIRECTORY            ==> C:\EDMMGR\LOG

EDM0604I 09:34 [ztoptask /065]  FLUSH_SIZE           ==> 1000

EDM0604I 09:34 [ztoptask /065]  THRESHHOLD           ==> -500000

EDM0604I 09:34 [ztoptask /065]  PIPE_SIZE            ==> 100000

EDM0604I 09:34 [ztoptask /065]  COLUMN_WIDTH         ==> 90

EDM1121I 09:34 [ztoptask /065]
======================================================================

EDM1121I 09:34 [ztoptask /065]  Profile Section(MGR_USERLOG)

EDM1121I 09:34 [ztoptask /065]
======================================================================

EDM0604I 09:34 [ztoptask /065]  ACTIVATE             ==> NO

EDM0604I 09:34 [ztoptask /065]  DIRECTORY            ==> C:\EDMMGR\LOG

EDM0604I 09:34 [ztoptask /065]  THRESHHOLD           ==> 500000

EDM0604I 09:34 [ztoptask /065]  FLUSH_SIZE           ==> 128

EDM0604I 09:34 [ztoptask /065]  COLUMN_WIDTH         ==> 128

EDM0604I 09:34 [ztoptask /065]  PIPE_SIZE            ==> 100000

EDM1121I 09:34 [ztoptask /065]
======================================================================

EDM1121I 09:34 [ztoptask /065]  Profile Section(MGR_DIRECTORIES)

EDM1121I 09:34 [ztoptask /065]
======================================================================

EDM0604I 09:34 [ztoptask /065]  DBPATH               ==> C:\EDMDB\

EDM0604I 09:34 [ztoptask /065]  REXX_PATH            ==> :\EDMMGR\REXX

EDM0604I 09:34 [ztoptask /065]  METHOD_PATH          ==> C:\EDMMGR\BIN

EDM1121I 09:34 [ztoptask /065]

======================================================================

EDM1121I 09:34 [ztoptask /065]  Profile Section(MGR_SIMULATION)

EDM1121I 09:34 [ztoptask /065]
======================================================================

EDM1121I 09:34 [ztoptask /065]
======================================================================
```

```
EDM1121I 09:34 [ztoptask /065]  Profile Section(MGR_METHODS)

EDM1121I 09:34 [ztoptask /065]
================================================================

EDM0604I 09:34 [ztoptask /065]  MSG_LIMIT            ==> 0

EDM0604I 09:34 [ztoptask /065]  LOG_LIMIT            ==> 0

EDM0604I 09:34 [ztoptask /065]  TIMEOUT              ==> 300

EDM1121I 09:34 [ztoptask /065]   ============================================

EDM1121I 09:34 [ztoptask /065]  Profile Section(MGR_CACHE)

EDM1121I 09:34 [ztoptask /065]
================================================================

EDM0604I 09:34 [ztoptask /065]  CACHE_SEGMENTS       ==> 2

EDM0604I 09:34 [ztoptask /065]  CACHE_SIZE           ==> 1048576

EDM0604I 09:34 [ztoptask /065]  CACHE_STATS          ==> NO

EDM0604I 09:34 [ztoptask /065]  NUM_OF_CACHE_ENTRIES ==> 1200

EDM1121I 09:34 [ztoptask /065]
================================================================

EDM1121I 09:34 [ztoptask /065]  Profile Section(MGR_TASK_LIMIT)

EDM1121I 09:34 [ztoptask /065]
================================================================

EDM0604I 09:34 [ztoptask /065]  TASKLIM_HARD         ==> 50

EDM0604I 09:34 [ztoptask /065]  TASKLIM_SOFT         ==> 40

EDM0604I 09:34 [ztoptask /065]  TASK_LOG_LIM         ==> 0

EDM0604I 09:34 [ztoptask /065]  TASK_RESO_LIM        ==> 64000

EDM0604I 09:34 [ztoptask /065]  TASK_STACK_SIZE      ==> 64000

EDM1121I 09:34 [ztoptask /065]
================================================================

EDM1121I 09:34 [ztoptask /065]  Profile Section(MGR_OBJECT_RESOLUTION)

EDM1121I 09:34 [ztoptask /065]
================================================================

EDM0604I 09:34 [ztoptask /065]  ALWAYS_CALL_ZADMIN   ==> YES

EDM1121I 09:34 [ztoptask /065]
================================================================

EDM1121I 09:34 [ztoptask /065]  Profile Section(MGR_SMTP_MAIL)

EDM1121I 09:34 [ztoptask /065]
================================================================

EDM0604I 09:34 [ztoptask /065]  RETRY_INTERVAL       ==> 300
```

```
EDM0604I 09:34 [ztoptask /065]  MAX_TIME_IN_SPOOL    ==> 4320

EDM1121I 09:34 [ztoptask /065]
================================================================

EDM0007I 09:34 [ztoptask /065]  Attaching Clock Manager Task

EDM0001I 09:34 [zclkmgr  /079]  Clock Manager Task has started

EDM0007I 09:34 [ztoptask /065]  Attaching Task Manager Task

EDM0001I 09:34 [ztaskmgr /082]  Task Manager Task has started

EDM0007I 09:34 [ztaskmgr /082]  Attaching [zutilmgr]

EDM0007I 09:34 [ztaskmgr /082]  Attaching [zrexxmgr]

EDM0007I 09:34 [ztaskmgr /082]  Attaching [ztcpmgr]

EDM0001I 09:34 [zutilmgr /069]  Utility Manager Task has started

EDM0001I 09:34 [ztcpmgr  /115]  TCP Manager Task has started

EDM0001I 09:34 [zrexxmgr /094]  REXX Manager Task has started

EDM0001I 09:34 [ztermtsk /090]  Client task started

EDM0777I 09:34 [ztermtsk /090]  Connected to Client

EDM0999I 09:34 [204.7.83.53 /090] Object not found - ZTEMP

EDM0777I 09:34 [204.7.83.53 /090] Object [EDMLOCTP] processing processing
                                  begins

EDM0777I 09:34 [204.7.83.53 /090] Object [EDMLOCTP] processing ends

EDM0777I 09:34 [204.7.83.53 /090] Object [ZCONFIG ] processing processing
                                  begins

EDM0532I 09:34 [204.7.83.53 /090] Method Method
                                  [C:\EDMMGR\BIN\EDMMPPRO.exe]
                                  started with task id=[328]

EDM0777I 09:34 [204.7.83.53 /090] Object [ZCONFIG ] processing ends

EDM0777I 09:34 [204.7.83.53 /090] Object [ZMASTER ] processing processing
                                  begins

EDM0999I 09:34 [204.7.83.53 /090] Object not found - ZSTATUS

EDM0532I 09:34 [204.7.83.53 /090] Method Method
                                  [C:\EDMMGR\BIN\EDMMEXIS.exe]
                                  started with task id=[344]

EDM0018E 09:34 [204.7.83.53 /090] EDMMEXIS STARTING
                                  PRIMARY.SYSTEMX.USER.USER1

EDM1181I 09:34 [204.7.83.53 /090] EDMMEXIS RECORD NOT FOUND
                                  PRIMARY.SYSTEMX.USER.USER1

EDM0532I 09:34 [204.7.83.53 /090] Method [C:\EDMMGR\BIN\ZEXIST.exe]
                                  started with task id=[372]

EDM0018E 09:34 [204.7.83.53 /090] EDMMEXIS STARTING
```

```
                                PRIMARY,SYSTEMX,USER,USER1

EDM1180I 09:34 [204.7.83.53 /090] EDMMEXIS RECORD WAS FOUND USER1

EDM0532I 09:34 [204.7.83.53 /090] Method Method
                                [C:\EDMMGR\BIN\EDMMPPRO.exe]
                                started with task id=[352]

EDM0000E 09:34 [204.7.83.53 /090] EDMMPPRO - OBJECT [ZXREF   ] NOT
                                FOUND

EDM0539E 09:34 [204.7.83.53 /090] SETTING OBJECT RESOLUTION STOP STOP
                                CODE

EDM0532I 09:34 [204.7.83.53 /090] Method C:\EDMMGR\BIN\EDMMPPRO.exe]
                                started with task id=[376]

EDM0532I 09:34 [204.7.83.53 /090] Method Method
                                [C:\EDMMGR\BIN\EDMMPPRO.exe]
                                started with task id=[336]

EDM0532I 09:34 [204.7.83.53 /090] Method Method
                                [C:\EDMMGR\BIN\EDMMPPRO.exe]

                                started with task id=[316]

EDM0532I 09:34 [204.7.83.53 /090] Method Method
                                [C:\EDMMGR\BIN\ZGETPROF.exe]

                                started with task id=[372]

EDM0595E 09:34 [204.7.83.53 /090] PROFILE ERROR LOCATING CLASS :

                                ZSTATUS_CLASS_

EDM0777I 09:34 [204.7.83.53 /090] Object [ZMASTER ] processing ends

EDM0777I 09:34 [204.7.83.53 /090] Object [ZREQSVCO] processing processing
                                begins

EDM0777I 09:34 [204.7.83.53 /090] Object [ZREQSVCO] processing ends

EDM0777I 09:34 [204.7.83.53 /090] Object [ZREQAUDO] processing processing
                                begins

EDM0388W 09:34 [204.7.83.53 /090] WARNING - INSTANCE
                                C:\EDMDB\PRIMARY\ZSYSTEM\ZPROCESS\

                                ZREQAUDO DOES NOT EXIST EDM2509E
                                09:34 [204.7.83.53 /090] FILE NOT

                                FOUND C:\EDMDB\PRIMARY\ZSYSTEM\

                                ZPROCESS\ZREQAUDO

EDM0013E 09:34 [204.7.83.53 /090] PUT OBJECT ZAUDITS-OBJECT NOT NOT
                                FOUND

EDM0777I 09:34 [204.7.83.53 /090] Object [ZREQAUDO] processing ends

EDM0777I 09:34 [204.7.83.53 /090] Object [ZCLIENT ] processing begins
```

```
                                  EDM0532I 09:34 [204.7.83.53 /090] ] Method
                                  [C:\EDMMGR\BIN\EDMMPPRO.exe]
                                  started with task id=[364]

EDM0777I 09:34 [204.7.83.53 /090] Object [ZCLIENT ] processing ends

EDM0999I 09:34 [204.7.83.53 /090] returned rc: 99 - Received end of
                                  session

EDM0532I 09:34 [204.7.83.53 /090] Method Method

                                  [C:\EDMMGR\BIN\ZPUTPROF.exe]
                                  started with task id=[336]

EDM0002I 09:35 [204.7.83.53 /090] Client task has ended

EDM0010I 09:35 [ztoptask /065]    Cmd Environment EDMLINK could not not
                                  be
                                  deleted

EDM0010E 09:35 [ztoptask /065]  REXX host cmd env cleanup completed
                                with errors

EDM0000I 09:35 [ztoptask /065]  Shutting Down EDM Manager [999]

EDM0000I 09:35 [ztoptask /065]  Terminating Task [ztcpmgr] ID=[115]

EDM0000I 09:35 [ztoptask /065]  Terminating Task [zrexxmgr] ID=[94]

EDM0000I 09:35 [ztoptask /065]  Terminating Task [zutilmgr] ID=[69]

EDM0000I 09:35 [ztoptask /065]  Terminating Task [ztaskmgr] ID=[82]

EDM0000I 09:35 [ztoptask /065]  Terminating Task [zclkmgr] ID=[79]

EDM0002I 09:35 [zlogmgr  /118]  EDM Manager terminating
```

6 EDM Manager Messages

This chapter contains the EDM Manager messages that are generated by the EDM Manager logging facility. The message descriptions explain the meaning of the message, and any system actions that may occur. The descriptions also suggest an action or actions (if any) that can be taken by the system administrator to diagnose and correct a problem.

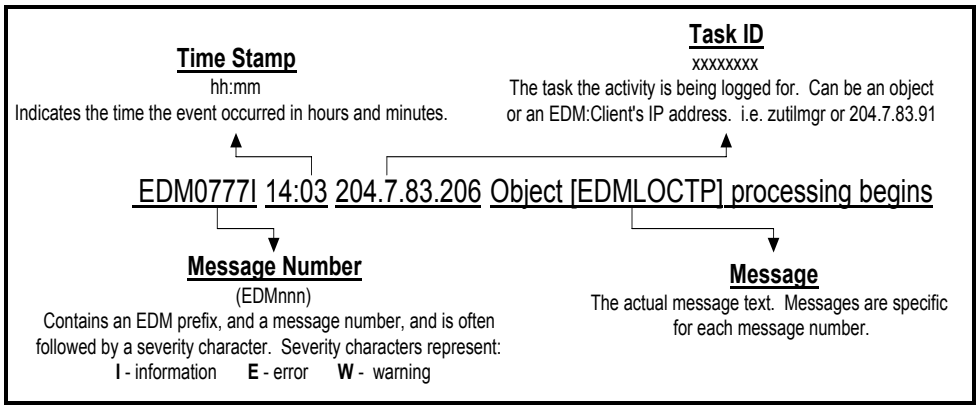
How to Use This Chapter

This chapter provides you with the information you need to interpret the messages that are written to the EDM Manager log.

The messages are written in numerical order. Read the following sections to understand how to use this chapter.

Description of the EDM Manager Message Format

All EDM Manager messages are preceded by an identifier and followed by the text of the message. The illustration below explains the format of an EDM Manager message.



Note: To help in problem determination, you might want to refer seeto Chapter 3,3*: "Tuning the EDM Manager for Windows NT."* The descriptions and examples of the EDMPROF.DAT settings in this chapter may provide greater insight into the EDM task named in the message text.

EDM Manager Messages

The EDM Manager message reference follows.

EDM0000I hh:mm taskid REFRESHING THE TIME

This message is produced every minute that the EDM Manager is operating.

EDM0001I hh:mm taskid *managertaskname* HAS STARTED

This message tells you that the EDM Manager task displayed in the message has started within the EDM Manager.

Each task provides a specific set of services. The following table lists and describes the possible values of *managertaskname***.**

The EDM Manager Task List

EDM Manager Task	Description
AGENT MANAGER TASK	Reserved for future use.
CLOCK MANAGER TASK	Wakes up every minute to update the time/date and flush the log.
LOG MANAGER TASK	Flushes the log messages periodically.
REXX MANAGER TASK	Runs REXX programs. A “hot” task that allows people to learn system REXXs that perform functions that persist in the environment (as opposed to transient methods, which do not persist).
TASK MANAGER TASK	Responsible for timing out tasks and restarting system tasks.
TCP/IP MANAGER EMULATOR TASK	Allows connections from TCP/IP EDM Clients.
TCP/IP NOTIFY MANAGER TASK	Notifies TCP/IP EDM Clients to connect.
CLIENT TASK	Actually initiates the EDM Client connect.

EDM0002I hh:mm taskid *managertaskname* HAS ENDED

This message tells you that the EDM Manager task displayed in the message has ended within the EDM Manager.

Each task provides a specific set of services. See Table 1the table above: *“The EDM Manager Task List”* for the possible values of *managertaskname***.**

EDM0007I hh:mm taskid ATTACHING *managertaskname* [*taskdescription*]

This message tells you that the EDM Manager task displayed in the message is being attached.

This message is written to the message log after the Clock Manager Task is started, and before the task actually starts.

EDM0010E hh:mm taskid *objectname* ERROR INITIALIZING HEAPS

This message tells you that the EDM Manager was not able to initialize the heaps for the *objectname* object. The reason for this error may be found in prior messages.

EDM0361E hh:mm taskid ERROR INITIALIZING OBJECT *objectname*

This message tells you that the EDM Manager was not able to initialize the *objectname* object. The reason for this error may be found in prior messages.

EDM0388W hh:mm taskid WARNING - INSTANCE *class.instance* DOES NOT EXIST

This message tells you that the instance *class.instance* was referenced during an object resolution, but the instance could not be found in the current configuration database (that is, in the Primary or Secondary file).

This could be the result of a missing or incorrectly spelled instance name, or a failure during variable substitution processing.

EDM0516W hh:mm taskid WARNING-RESOURCE SIZE (ZRSCSIZE) NOT NUMERIC

This message tells you that the value specified for the resource is not numeric. This may impede further processing.

EDM0532E hh:mm taskid *path* NO SUCH FILE OR DIRECTORY

This message tells you that the EDM Manager is not able to find the file or directory listed in the path.

EDM0532I hh:mm taskid CURRENT PATH *path*

This message tells you from where the EDM Manager is currently operating or retrieving a file.

EDM0532I hh:mm taskid *methodname* METHOD ID

This message identifies the method the EDM Manager is currently running.

EDM0566E hh:mm taskid *classname* CLASS DOES NOT CONTAIN CONTROL INFORMATION]

This message tells you that the class cited in the message does not have associated control information for the entire class.

EDM0579E hh:mm taskid - OBJECT RESOLUTION FAILURE FOR OBJECT *objectname.variable*

This message tells you that the EDM Manager was not able to resolve the *objectname.varaiable* object. The reason for this error may be found in prior messages.

EDM0585E hh:mm taskid -ERROR RETRIEVING HEAP *nn* FOR OBJECT *objectname*

This message tells you that the EDM Manager was not able to initialize the specific heap for the *objectname* object.

EDM0604I hh:mm taskid - *sectionname -> value*

This message tells you the individual values for the settings of the EDM profile, which are found in the EDMPROF.DAT file.

EDM0777I hh:mm taskid - OBJECT *objectname* PROCESSING BEGINS

This message tells you that the EDM Manager has begun the object resolution process for the *objectname* object.

EDM0777I hh:mm taskid - OBJECT *objectname* PROCESSING ENDS

This message tells you that the EDM Manager has finished the object resolution process for the *objectname* object.

EDM0777I hh:mm taskid - CONNECTED TO CLIENT *clientname*

This message tells you that the EDM Manager has connected to the EDM Client *clientname*.

EDM0777E hh:mm taskid - INSUFFICIENT MEMORY SIZE *path*

This message tells you that there was not sufficient memory in the path name *path* to perform a specific operation.

EDM0999I hh:mm taskid - RETURNED RC:99 END OF SESSION

This message tells you that the EDM Manager has received the return code that terminates communications between the EDM Manager and an EDM Client. This in effect ends an EDM Client connect.

EDM0999E hh:mm taskid - OBJECT NOT FOUND [*address*]

This message tells you that the EDM Manager was not able to find an expected object at the address cited.

EDM1121I hh:mm taskid - PROFILE SECTION *sectionname*

This message tells you the section of the Profile file that the EDM Manager is currently processing.

EDM2500E hh:mm taskid - UNABLE TO PACK OBJECTID

This message lets you that the EDM Manager was not able to pack the unique object identifier (OBJECTID) for a specific object.

EDM2509E hh:mm taskid - FILE NOT FOUND *filename*

This message tells you that the EDM Manager was not able to find the expected *filename* file.

EDM2533E hh:mm taskid - ERROR ALLOCATING HEAP *objectname*

This message tells you that the EDM Manager was not able to allocate a heap for the *objectname* object.

EDM3000E hh:mm task id NO HEAPS WERE ALLOCATED FOR OBJECT *objectname*

This message tells you that the EDM Manager did not allocate any occurrences for the *objectname* object.

7 Applying Windows NT Security to EDM

This chapter explains how to use Windows NT security with your EDM Enterprise Manager. To do this, you must:

- Define EDM IDs and passwords to Windows NT security.
- Configure EDM to call the EDMMSGNR method so it will invoke the Windows NT security program.

EDMMSGNR Will Permit EDM to use the Security Overview

Administrators and Users are connected to Windows NT security using the same basic procedure. A brief outline of this procedure is given below.

➢ **To Connect Administrators and Users to Windows NT Security:**

1. Specify EDM ID and Password information to Windows NT security.
2. Create an instance in the ZMETHOD class of the ZSYSTEM domain, and connect it to EDMMSGNR.
3. Connect your ZMETHOD instance to an instance of the ZPROCESS class in the ZSYSTEM domain.

The following table will help you locate the information you need to complete the process outlined above.

If You Want To Learn How To:	Turn To The Section:
Define EDM Administrator ID, User ID and Password information to Windows NT security	Defining EDM to Windows NT Security
Storing the Windows NT Security Object	Storing the Security Object
Using NT Workgroups in EDM	Referencing the Security Object

Defining EDM to Windows NT Security

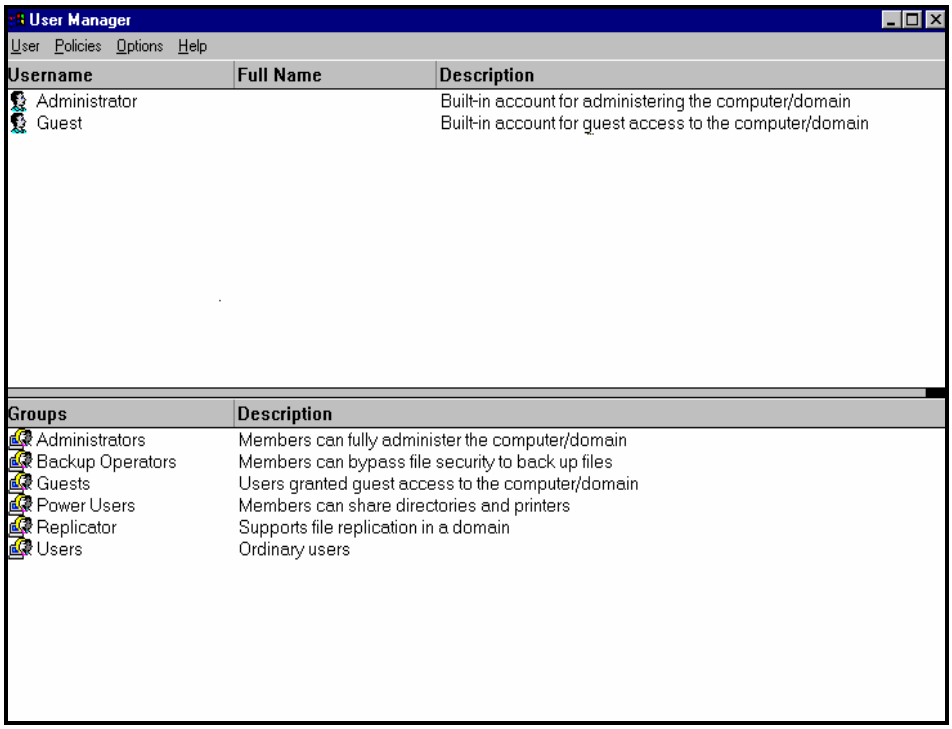
If you are currently using Windows NT, you may want to use its security program with your Manager. To do so, you will need to connect your Windows NT security configuration to EDM.

The first task you will need to perform is to define your EDM Administrator ID, User ID and password information to Windows NT security. This must be done on your Manager.

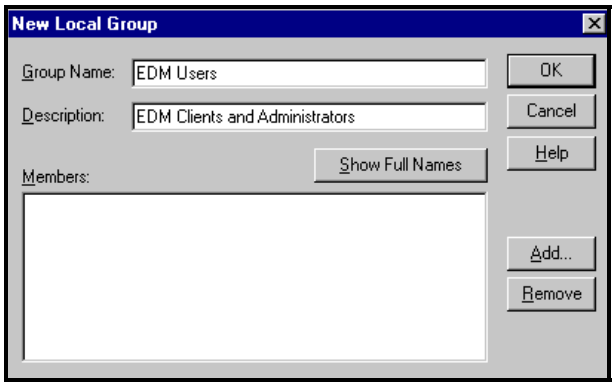
Before you begin defining EDM information to Windows NT security, follow the procedure below to create a new security group that your EDM IDs will be connected to.

➢ **To Create the Security Group for EDM IDs in Windows NT Security:**

1 Open the security program by choosing **Administrative Tools (Common)**, then **User Manager** from the Windows taskbar. The following window is displayed.



2 Select **New Local Group** from the **User** menu. The following dialog box is displayed.



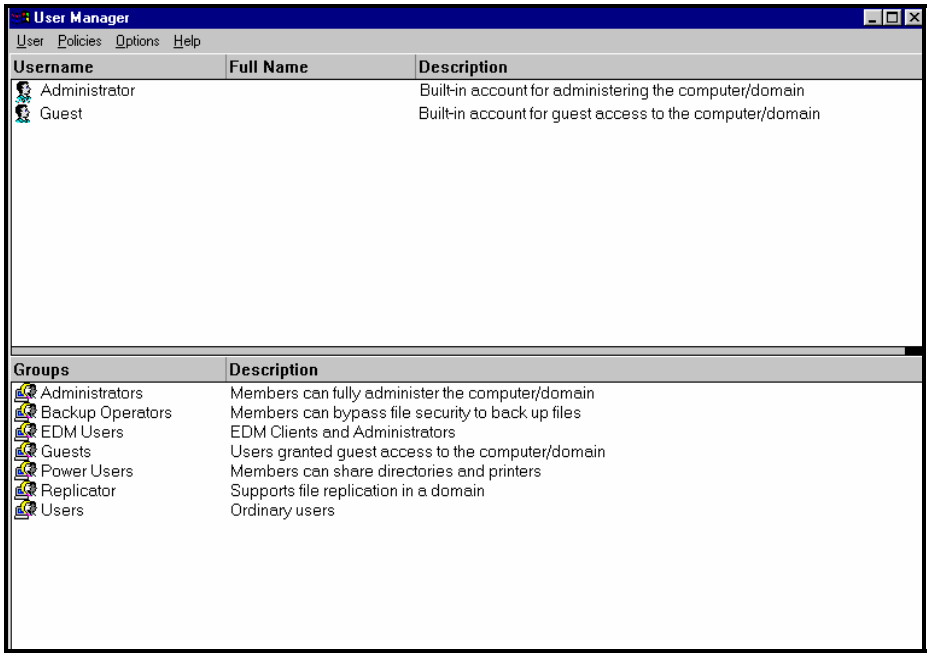
Type `EDM Users` in the **Group Name** text box. Type `EDM Clients and Administrators` in the **Description** text box. Choose **OK**. You will return to the **User Manager** window.

IDs for EDM Clients and Administrators will be connected to this group.

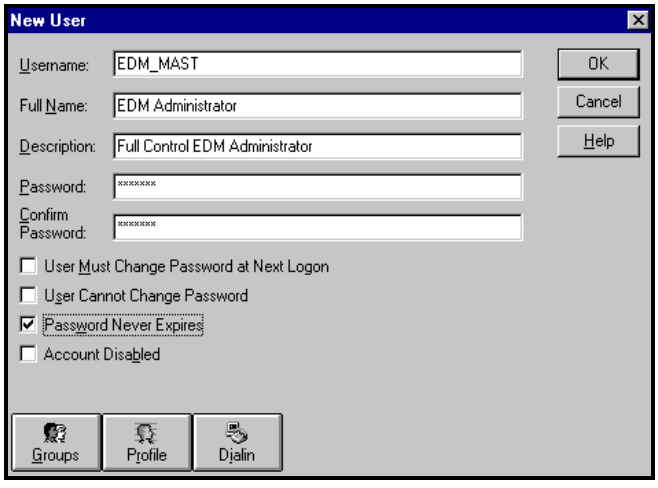
You will need to perform the procedure below for each EDM Administrator ID, and User ID you want to define to Windows NT security.

➢ **To Define EDM ID and Password Information to Windows NT Security:**

1 Open the security program by choosing **Administrative Tools (Common),** then **User Manager** from the Windows taskbar. The following window is displayed.



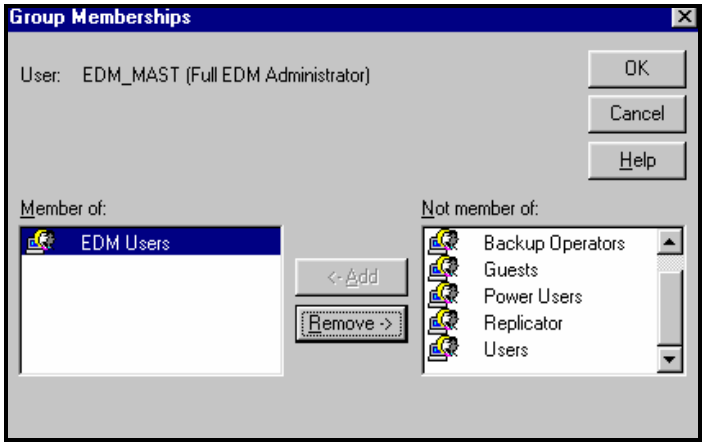
2 Select **New User** from the **User** menu. The following dialog box is displayed.



Type either the Administrator ID or User ID in the **Username** text box. Type the password information in the **Password** and **Confirm Password** text boxes then, select the **Password Never Expires** check box. Type the name for this ID in the **Full Name** text box.

Optionally, you can specify information in the **Description** text box. This information will enable you to provide a more intuitive description for the user.

3 Choose the **Groups** button. The following dialog box is displayed.

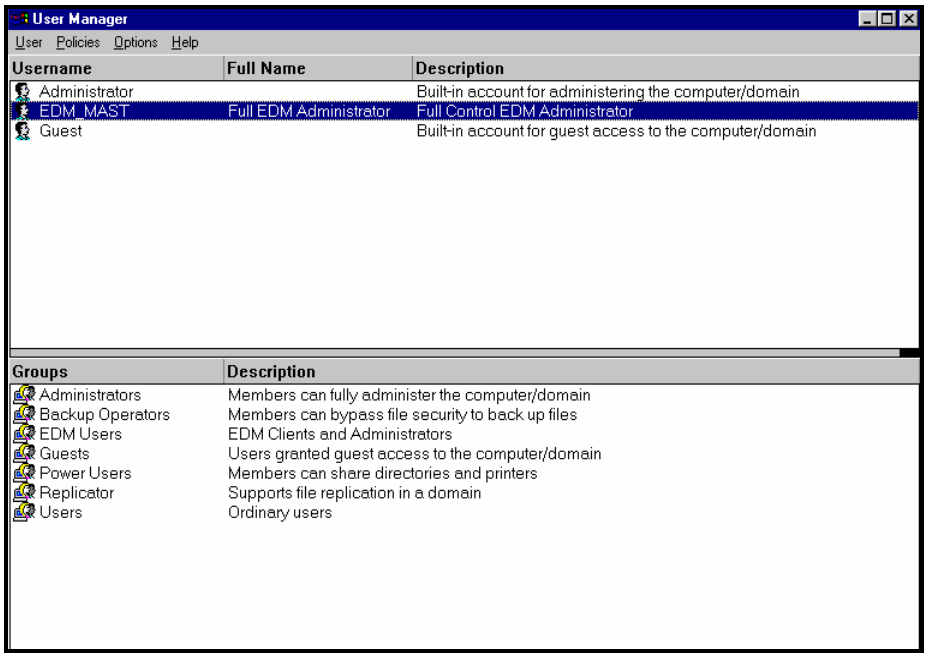


Add the groups you want this ID to be connected to by highlighting the group in the **Not member of:** list box and selecting **<-Add**. In the example above, the IDs have been connected in the EDM Users group that was previously created.

Note: Do not connect EDM Clients and EDM Administrators to groups that give them the ability to log on to the server itself (i.e. the Users group).

To remove a group from this ID, highlight it in the **Member of:** list box, and select **Remove->.** When you have finished, choose **OK**. The User Properties dialog box will be displayed again.

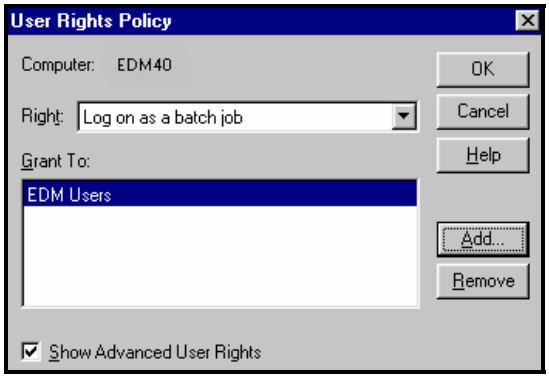
4 Choose **OK** to return to the **User Manager** window. You will see the new ID you just added displayed in the **Username** column.



In the **User Manager** window, select **User Rights Policy** from the **Policies** menu. The **User Rights Policy** dialog box is displayed.

If you are defining a typical EDM ID to Windows NT security, see step **5**. If you are defining an ID that will have the ability to start and stop the Manager, see step **6**.

5 After selecting **User Rights Policy** fFrom the **Policies** menu, select **User Rights Policy**. yYou will see the dialog box below.



Choose **OK** to save your selection.

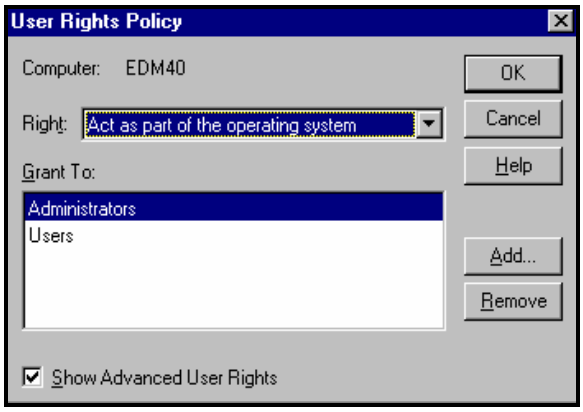
Select the **Right** drop down list box, and select the **Log on as a batch job** option. This will allow the Manager to impersonate an ID so it can retrieve information from the operating system. This right is required to be specified for only one of the groups an ID is connected to.

To ensure that the ID is connected to a group with this right, you can create a specific group that is connected to the right: **Log on as batch job**.

The **Grrant To** list box displays the groups the user is connected to. Verify that the user is connected to the correct groups. If you need to adjust the connections, use the **Add...** and **Remove** option buttons.

At the bottom of this screen, you see the check box **Show Advanced User Rights**. You must select this check box before continuing. Choose **OK**.

6 After selecting **User Rights Policy** f From the **Policies** menu, select **User Rights Policy**. You will see the dialog box belowrom the **Policies** menu, you will see the dialog box below.



Choose **OK** to save your selection.

Select the **Right** drop down list box, and select the **Act as part of the operating system** option. This will allow the ID you have associated with starting and stopping the Manager to retrieve information from the operating system.

Note: When you run the EDM Manager as a service, you must give the NT Administrator group the following right: **Act as part of the operating system**.

The **Grant To** list box displays the groups the user is connected to. Verify that the user is connected to the correct groups. If you need to adjust the connections, use the **Add...** and **Remove** option buttons.

Note: The **Grant To** list box should display the group the userid, (the ID that is used to start the EDM Manager itself) belongs to. If the group for the userid is not in the dialog box, add it using the procedure described above.

At the bottom of this screen, you see the check box **Show Advanced User Rights**. You must select this check box before continuing. Choose **OK**.

Connection Overview

Once you have set up Windows NT security, and have defined EDM Administrator IDs, User IDs, and passwords to it, you can connect it to EDM using the EDMMSGNR method.

Invoking the use of Windows NT security will involve attaching the EDMMSGNR method to a ZMETHOD class instance. You will create the ZMETHOD instances you need by copying existing instances from within the ZMETHOD class.

The instances you will copy are the EDM_ZADMIN_ SIGNON and EDM_SIGNON instances. These instances are used to configure EDM to use its own security program through the use of the EDMSIGN method. Both are located in the ZSYSTEM domain's ZMEHTOD class. This domain is located in the PRIMARY file.implement password security for your users by invoking the EDMMSGNR method.

The following table summarizes the procedure used to connect Administrators and Users.

Step	For Administrators	For Users
1	Go to the ZSYSTEM domain, ZPROCESS class. Select the ZADMIN instance. In the first 'METHOD' field, enter ZSYSTEM.ZMETHOD.EDM_ADMIN_ EXTENDED_SIGNON.Copy the EDM_ZADMIN_SIGNON instance and call it NT_ADMIN_SIGNON.	Copy the EDM_SIGNON instance and call it NT_CLIENT_SIGNONGo to the ZSYSTEM domain, ZPROCESS class. Select the ZMASTER instance. In the first 'METHOD' field, enter ZSYSTEM.ZMETHOD.EDM_CLIENT_EXTENDED_SIGNON.
2	Connect the NT_ADMIN_SIGNON instance toEnsure that a "Connect To' field in the ZADMIN instance exists: of the ZPROCESS class.ZACCESS.ADMINID.&(ZADMIN. ZUSERID)	Ensure that a "Connect To' field in the ZMASTER instance exists: SYSTEMX.USER.&(ZMASTER. ZUSERID)(EDMSETUP)Connect the NT_CLIENT_SIGNON instance to the ZMASTER instance of the ZPROCESS class.

The ZPROCESS class is in the PRIMARY file's ZSYSTEM domain.

When you connect the EDMMSGNR method to the instance you created in the ZMETHOD class, you are telling the instance to execute the EDMMSGNR method when it is resolved.

The following section takes you through the process of connecting your ZMETHOD instance to the ZADMIN/ZMASTER instance of the ZPROCESS domain. By making this connection, you are instructing the ZADMIN/ZMASTER object to read the ZMETHOD instance during the EDM object resolution process. Because this instance is connected to EDMMSGNR, it will execute the method.

Before you begin connecting the EDMMSGNR method for Administrators, you should check your edmprof.dat file to ensure that the Access parameter is set correctly to ADMIN=ALLOW.

Example

```
[MGR_ACCESS]

ADMIN = ALLOW
```

> **Note**: Before you begin, you should backup your ZMASTER and ZADMIN objects so you can retrieve them if problems occur with their security configurations.

Warning: The Access section of the edmprof.dat file must remain ADMIN=ALLOW. If you change this option to DENY, you will not be able to use the security configuration you have constructed in Windows NT because you will be denied access to your EDM Manager.

You must also check your PROFILE to see that a previous ZADMIN object has not been stored there. If there is a ZADMIN object already present, delete the instance in the PROFILE.

The ZADMIN object's corresponding instance can be found by looking in the domain of the PROFILE file that is identified by the user's ID. Open the ZADMIN class from within this domain. The OBJECT instance is the instance that should be deleted if you are about to define the user to Windows NT security for the first time.

Invoking the EDMMSGNR Method for Administrators and Users

This section will take you through a step-by-step summary of the process you will use to invoke the use of Windows NT security while running EDM.

To use Windows NT security, you will need to connect the EDMMSGNR method to the NT_ADMIN_SIGNON instance (Administrators) or the NT_CLIENT_SIGNON instance (Users) in the ZMETHOD class.

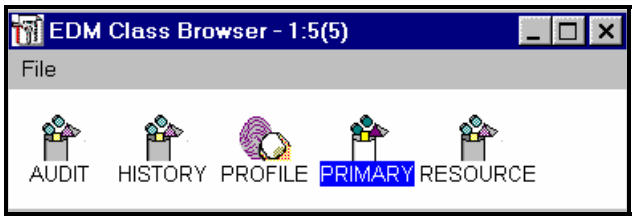
You must also connect this instance the ZADMIN instance (Administrators) or the ZMASTER instance (Users). The ZADMIN instance is located in the ZPROCESS class of the ZSYSTEM domain.

Connecting EDMMSGNR to the NT_ADMIN_SIGNON or NT_CLIENT_SIGNON Instance

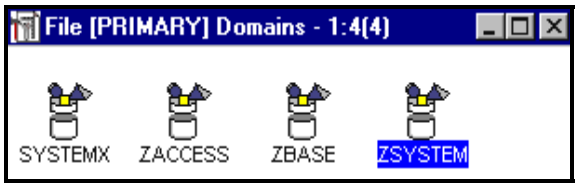
The first step in connecting Windows NT security to EDM is to connect EDMMSGNR to your ZMETHOD class instance. This will instruct the instance to execute the EDMMSGNR method whenever it is used. EDMMSGNR will invoke Windows NT security during the EDM object resolution process.

➢ **To Connect EDMMSGNR to the NT_SIGNON or NT_CLIENT_SIGNON Instance:**

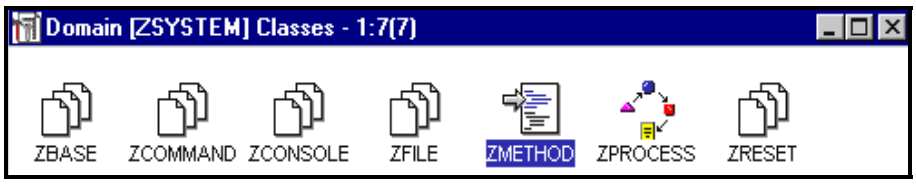
1. Select **EDM Administrator** from the Windows taskbar.
2. Select **EDM Class Browser** to connect to the Manager. The following window is displayed.



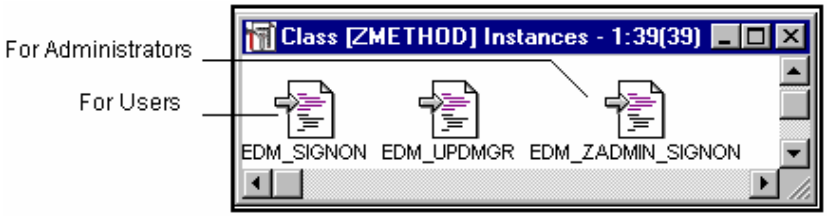
3. Choose the **PRIMARY** file. The following window is displayed.



4. Select the **ZSYSTEM** domain. The following window is displayed.



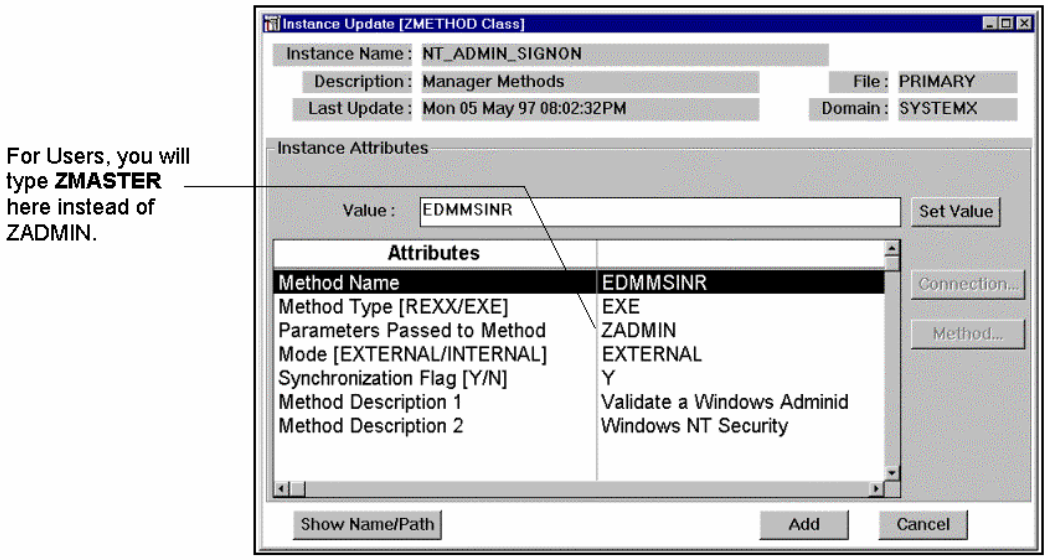
5. Select the **ZMETHOD** class. The following window is displayed.



6 Depending on whether you are performing the connection for Administrators or Users, you will do the following.

For	Do the following
Administrators	Copy the EDM_ZADMIN_SIGNON instance that is installed with EDM, and call it **NT_ADMIN_SIGNON**.
Users	Copy the EDM _SIGNON instance that is installed with EDM, and call it **NT_CLIENT_SIGNON**.

You can copy an instance by right clicking on it and selecting **Copy Instance** from the drop down menu that appears. The following window is displayed.



From within the instance, perform the procedures described in the following table.

For Administrators (NT_ADMIN_SIGNON instance)

Highlight this attribute value:	In the Value text box type:
Member Name of Method	EDMMSGNR, then choose **Set Value**.
Parameters Passed to Method	ZADMIN, then choose **Set Value**. This specifies that the parameters passed to method attribute be set to ZADMIN.

For Users (NT_CLIENT_SIGNON instance)

Highlight this attribute value:	In the Value text box type:
Member Name of Method	EDMMSGNR, then choose **Set Value**.
Parameters Passed to Method	ZMASTER, then choose **Set Value**. This specifies that the parameters passed to method attribute be set to ZMASTER.

When you are finished entering the attribute values, select **Add** to accept the changes you have made to the instance. To close the instance without changing it, choose **Cancel**.

By entering the information above, you are connecting the EDMMSGNR method to the NT_ADMIN_SIGNON instance (or NT_CLIENT_SIGNON instance). Now, whenever this instance is called, it will tell the EDMMSGNR method to look at the ZADMIN object (or ZMASTER object) for user ID and password verification.

When your ZADMIN object points to the NT_ADMIN_SIGNON instance during the EDM object resolution process, the instance will execute the EDMMSGNR security method. This method enables EDM to use the Windows NT security program.

Connecting the NT_ADMIN_SIGNON/ NT_CLIENT_SIGNON Instance to ZADMIN/ZMASTER

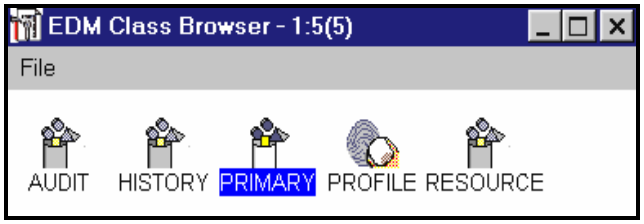
Now that EDMMSGNR is connected to the NT_ADMIN_SIGNON (or NT_CLIENT_SIGNON) instance, you are ready to connect the instance, of the ZMETHOD class, to the ZADMIN (or ZMASTER) instance of the ZPROCESS class.

When you connect the NT_ADMIN_SIGNON/NT_CLIENT_SIGNON instance, it is important that the first attribute in the ZADMIN/ZMASTER instance is connected to it. If it is not, other tasks will be performed before the EDMMSGNR method is invoked. This will undermine your security set up by allowing users to enter applications before you have implemented a security program.

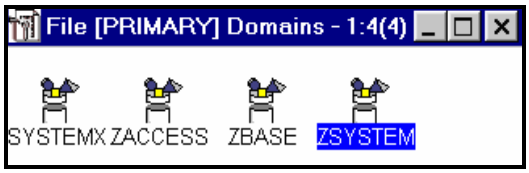
➢ **To Connect the NT_ADMIN_SIGNON/ NT_CLIENT_SIGNON Instance to ZADMIN/ZMASTER:**

1 Select the **EDM Administrator** from the Windows taskbar.

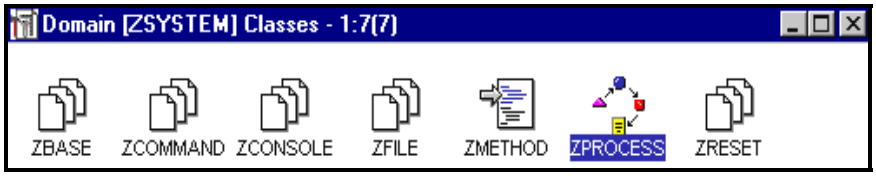
2 Select the **EDM Class Browser**. The following window is displayed.



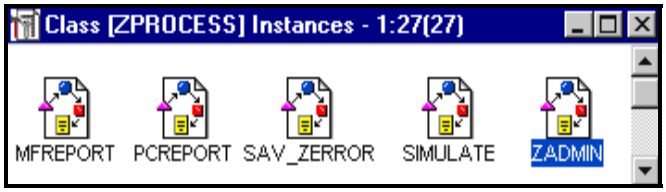
3 Select the **PRIMARY** file. The following window is displayed.



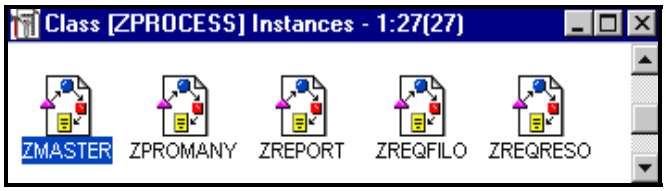
4 Choose the **ZSYSTEM** domain. The following window is displayed.



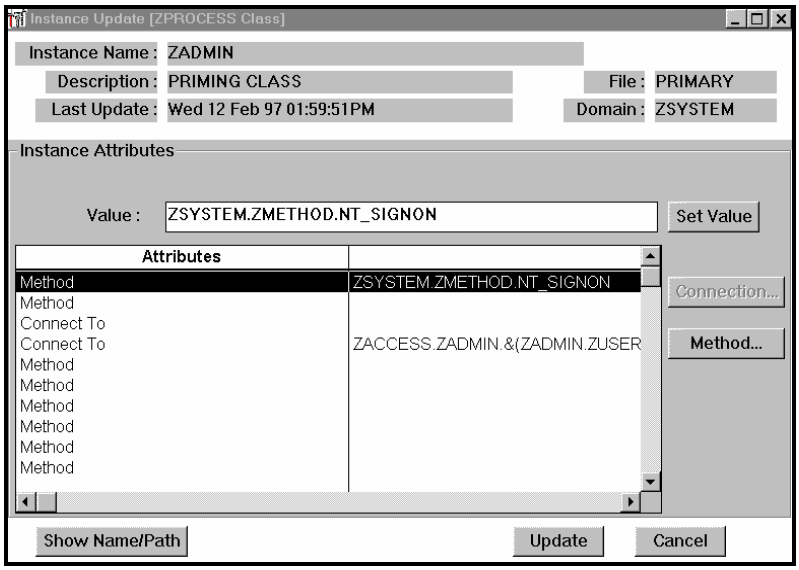
5 Choose the **ZPROCESS** class. The following window is displayed.



For Users



6 To connect Administrators, choose the ZADMIN instance. For Users, choose the ZMASTER instance. The following window is displayed.



7 Highlight the first attribute value and type the following information in the **Value** text box.:

For:	Type:
Administrators	ZSYSTEM.ZMETHOD.NT_ADMIN_SIGNON
Users	ZSYSTEM.ZMETHOD.NT_CLIENT_SIGNON

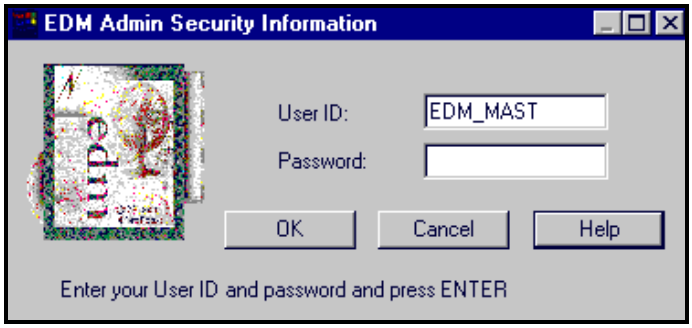
Select **Set Value** to accept the new attribute value. Choose the **Update** option to save the changes to the ZADMIN instance. To exit the instance without making any changes, select the **Cancel** option.

Note: The NT_ADMIN_SIGNON instance must be connected to the first attribute value in the ZADMIN instance.
The NT_CLIENT_SIGNON instance must be connected to the first attribute value in the ZMASTER instance.

By performing the preceding steps, you have effectively forced the ZADMIN/ZMASTER instance, of the ZPROCESS class, to point to your ZMETHOD class instance. This instance executes the EDMMSGNR method that will invoke Windows NT security.

Defining Administrator and Password Information to EDM

As in Windows NT, you will need to define your administrator's ID and password information to EDM. This is a relatively simple process, and is completed by submitting the information when it is requested in the EDM Admin Security Information panel. The EDM Admin Security Information dialog box is shown below.



When defining administrator ID and password information for the first time, type the administrator's ID in the **User ID:** text box. Type the administrator's password in the **Password:** text box.

Choose **OK** to enter the EDM Manager Database.

The information submitted in the signon panel will be stored in the Profile file in the EDM database. Each domain in the Profile file represents a userid. When an administrator signs on to the EDM Class Browser for the first time, a domain, representing the administrator's userid, is created in the Profile file.

The Windows NT Security Object

When EDMMSGNR is executed, as a result of the object resolution process, it will receive two variables from the inbound object to verify a user's logon rights. The two variables it receives are:

- ZUSERID User id
- ZPWD Password

If the two variables are not found in the inbound object, the process will be terminated, and an error will be sent back to the user.

If the variables are received, they will be checked against the security system. If they match, EDMMSGNR will impersonate the ID in order to obtain additional information from the operating system.

EDMMSGNR can impersonate an ID only if that ID is connected to a group that has been granted the right: **Log on as a batch job**. This right is granted to a group from the **User Rights Policy** dialog box in Windows NT security.

The procedure used to grant this right to a group is described in the section "*Defining EDM to Windows NT Security*" " aboveon page 163in this Chapter.

Whether the password verification process was a success or a failure, a security object called ZNTSEC will be created. The table below lists the variables for this object, and provides a brief description of each.

The ZNTSEC object is a temporary (in-storage) object that exists when there is a connection between an Administrator or User ID, and the Manager.

Variables for ZNTSEC

Variable	Description
ZNTUSER	The ID supplied in the inbound object.
ZMMSG	The message from the method.
ZMRC	The return code from the method.
ZNTGRPnn	The name of the group to which an ID belongs to. This variable will be created for each group the ID belongs to.
ZNTDMnn	The name of the domain to which the ID's group belongs to. This variable will be created for each group the ID belongs to.

The instorage object may have more than one ZNTGRP or ZNTDM variable. This is noted by the nn suffix at the end of variable name in the table above. A variable will be created for each group and domain an ID is connected to.

> **Note**: The last two variables in the ZNTSEC object will only be created if RC=EDM_METHOD_OK.

Two of the variables listed in the table above are also written to the inbound object, which can be viewed with the EDM Object Editor. The variables that are written to the inbound object are ZMMSG and ZMRC. For Administrators, this object is the ZADMIN object. For Users, the inbound object is the ZMASTER object.

The following messages are created from EDMMSGNR and are stored in the ZMMSG variable of the ZNTSEC object:

- User information put in the object
- Error obtaining user information
- Unable to Impersonated Logged On User
- Can not retrieve user info – no [LOG ON AS BATCH JOB] privilege is granted to user

The following table lists the return codes for EDMMSGNR, with a brief description of each. The return codes are stored in the ZMRC variable of the ZNTSEC object.

Return Codes for EDMMSGNR

Return Code	Description
0	EDM_METHOD_OK
4	EDM_METHOD_WARNING
16	EDM__METHOD_FAILURE

Storing the Security Object

Administrators can store a permanent record of the security object in the EDM database. By doing so, they can reference the access policies of an ID even when a communications link between the ID and the Manager does not exist.

There are two procedures that can be used to store this object, both use the Manager method EDMMPPRO. You can use EDMMPPRO to store the object by:

- Executing the EDMMPPRO on the instorage object from the command prompt.
- Creating an instance in the ZMETHOD class for EDMMPPRO, and connecting it to the ZPROCESS.ZMASTER instance. This automates the first procedure.

Manual Storage

If you want to manually store the security object in the database at selective times, you can execute EDMMPPRO on the object at any time from a command prompt. Although the use of EDMMPPRO can be automated, there are certain situations that will require you to employ the method manually, for example:

- If you do not want to store the security object on a routine basis because it information will only be needed sporadically, you will not automate its usage.
- If you need to update or replace the object stored in the Profile with a newer copy of the object, you will need the ability to execute the method manually.

To execute EDMMPPRO on the in-storage security object, type the following statement from the command prompt:

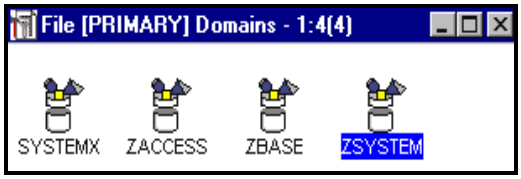
```
EDMLINK EDMMPPRO 'ZNTSEC'
```

Automated Storage

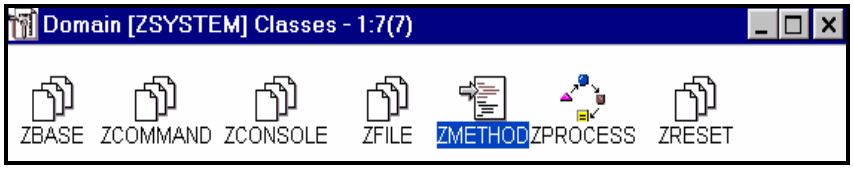
You can automate the process of using EDMMPPRO by creating an instance for it in the ZMETHOD class, and connecting the ZPROCESS.ZMASTER class instance to it. This will store a security object for each ID when it is created during a communications session with the Manager.

➢ **To create an EDMMPPRO instance:**

1. Select the **EDM Administrator** from the Windows taskbar.
2. Select the **EDM Class Browser** to connect to the Manager.
3. Choose the **PRIMARY** file from the screen that appears. The following window is displayed.

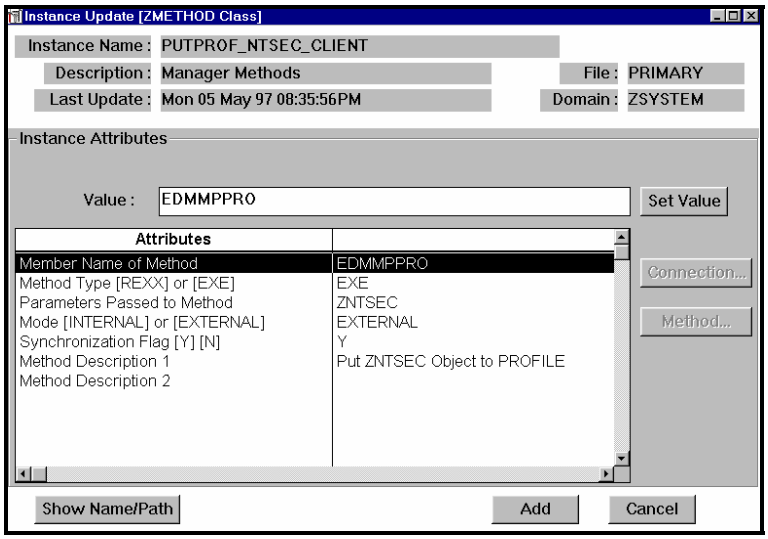


4. Select the **ZSYSTEM** domain. The following window is displayed.



5 Select the **ZMETHOD** class.

6 Copy the base instance from the class, and call it EDMMPPRO_NTSEC_CLIENT. You can copy an instance by right clicking on it and selecting **Copy Instance**. The following window is displayed.



In the EDMMPPRO instance:

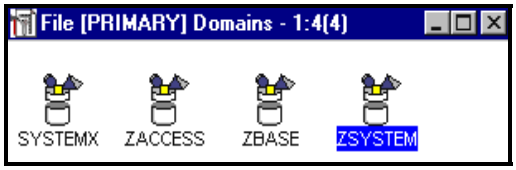
Select this attribute value and	In the Value text box type:
Member Name of Method	EDMMPPRO, then choose **Set Value**.
Method Type	EXE, then choose **Set Value**.
Parameters Passed to Method	ZNTSEC, then choose **Set Value**. This specifies that the parameters passed to method attribute be set to ZNTSEC.
Mode [INTERNAL] or [EXTERNAL]	EXTERNAL, then choose **Set Value**.
Method Description 1 (Optional)	Type a description for the method, then choose **Set Value**.

When you are finished entering the attribute values, select **Add** to add the new instance.

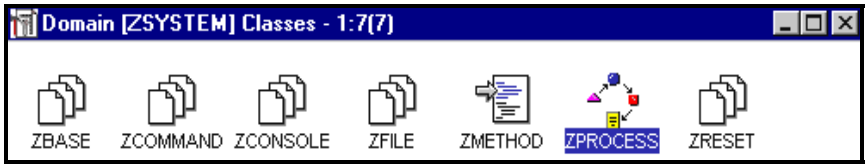
➢ **Next, you will connect the necessary ZPROCCES class instance to the EDMMPPRO method.**

1 To connect ZMASTER to EDMMPPRO:

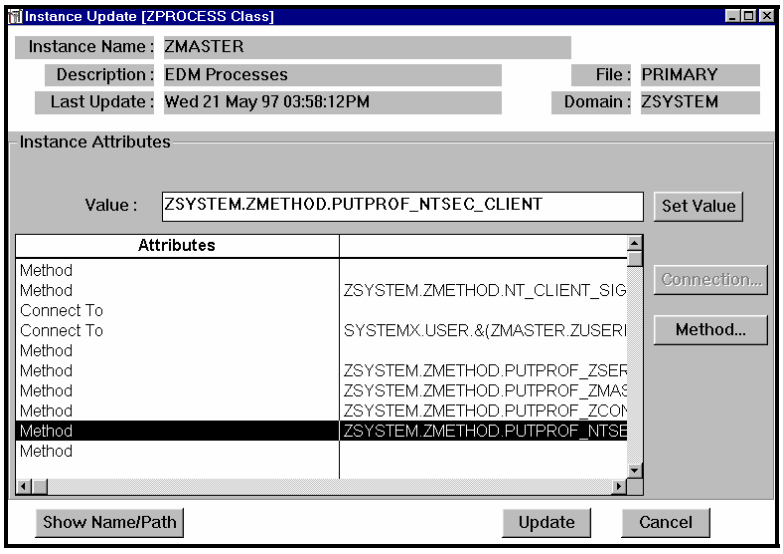
2 From the **EDM Class Browser** window, open the **Primary** file. The following window is displayed.



3 Choose the **ZSYSTEM** domain. The following window is displayed.



4 Choose the ZPROCCESS class, and open the ZMASTER class instance. The ZMASTER class instance is shown belowin the **Instance Update** window below.



Highlight a **Method** attribute that is listed below the one you used to connect to the EDMMSGNR method. Type the following statement in the **Value** text box:

```
ZSYSTEM.ZMETHOD.EDMMPPRO_NTSEC_CLIENT
```

Select **Set Value** to accept the new attribute value. Choose the **Update** option to save the changes to the instance.

The EDMMPPRO method will be executed each time a communications session is established with the Manager, and the ZNTSEC security object will be stored in the Profile file.

Note: The ZNTSEC object will be updated each time a connection is established with the EDM Manager.

Referencing the Security Object

The information from the ZNTSEC security object can be referenced by the SYSTEMX domain. This will place your userids into groups.

You must create an NTSGRP class in the SYSTEMX domain for the group instances that will be created. A class instance in the ZNTSGRP class will be created for each ZNTGRP variable in the ZNTSEC object. Each variable represents a group in the Windows NT security schema.

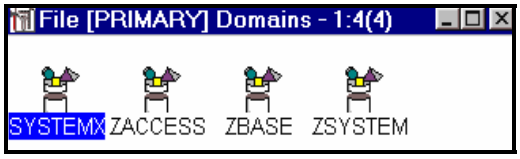
You must also connect the ZPROCESS.ZMASTER instance to each ZNTSEC group. If the group is not configured, it will not be represented in the SYTEMX domain.

Creating NTSGRP Classes in the SYSTEMX Domain

To use the information from the NTSEC security object in the SYSTEMX domain, you must add a class to the domain.

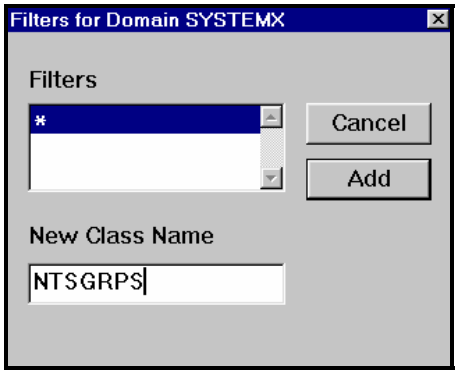
➢ **To add a class to the SYSTEMX domain:**

1 From the **EDM Class Browser** window, choose the **Primary** file. The following window is displayed.

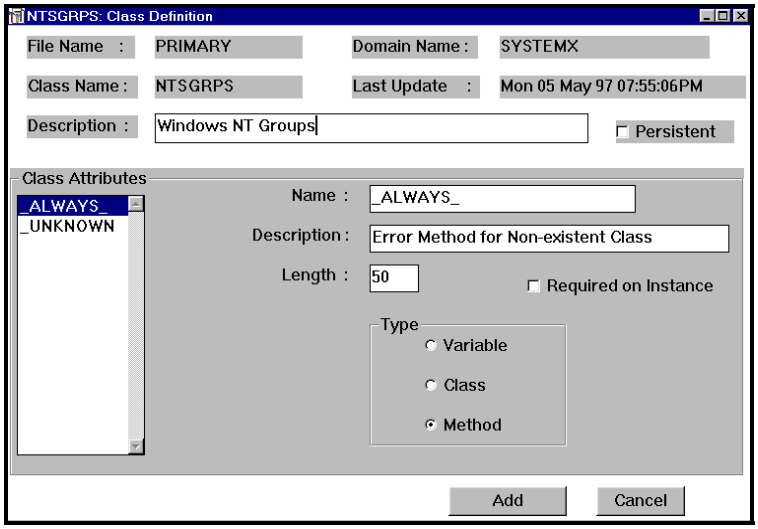


2 From the Primary File window, choose the **SYSTEMX** domain.

3 Choose the WORKGRP class with your right mouse button, and select **Copy Class**… from the drop down menu that appears. The following window dialog box appears.



4 In the **New Class Name** text box, type NTSGRPS. The following window is displayed.



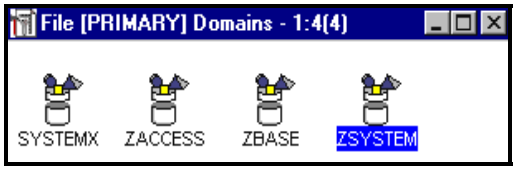
5 In the **Description** text box, type `Windows NT Groups`.

Connecting the ZPROCESS.ZMASTER Instance to the NTSGRP Classes

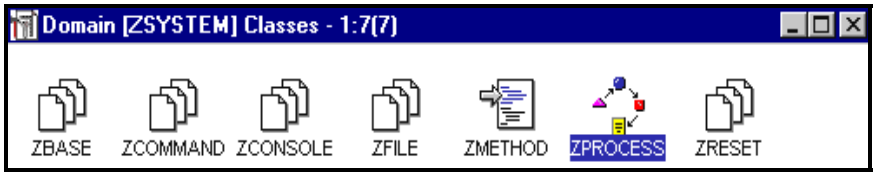
In order for you to connect the ZMASTER instance to the NTSGRP classes, you must add several Connect To attributes to the ZPROCESS class template. This will give the ZMASTER instance the extra fields it needs to connect to all of the NTSGRP classes.

➢ **To add Connect To attributes to the ZPROCESS class:**

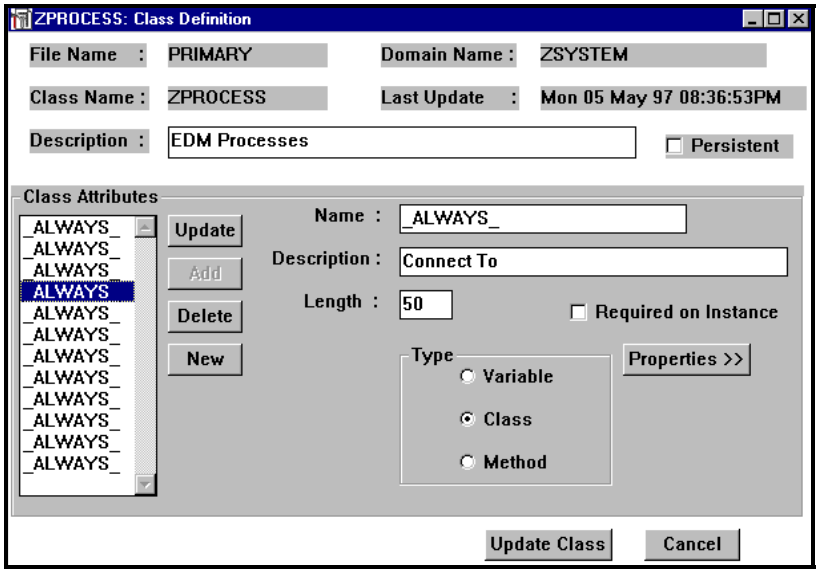
1 Choose the **PRIMARY** file. The following window is displayed.



2 Choose the **ZSYSTEM** domain. The following window is displayed.



3 Select the **ZPROCESS** class with your right mouse button, and choose **Update** from the drop down menu that appears. The following window is displayed.



4 Select **New**. The text boxes on the screen will be empty. Fill in the following information in the proper locations.

In the:	Do this
Name: text box	type `_ALWAYS_`
Description: text box	type `Connect To`
Length: text box	type `50`
Type group box	select the **Class** radio button

When you have specified the information above, choose **Add**.

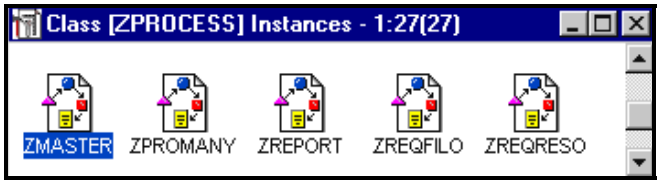
5 Repeat step **4** to add a Connect To attribute for each NTSGRP class you will have.

6 Select **Update Class** to save your changes.

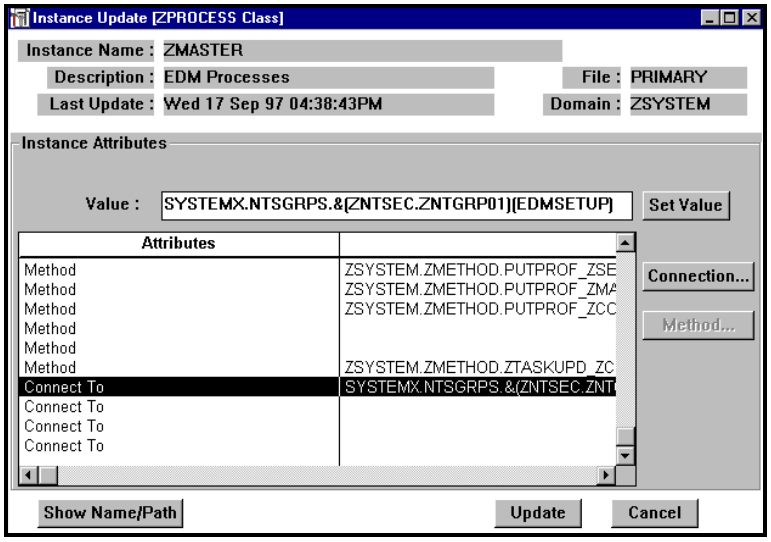
Now that you have added your Connect To attributes, you can connect the ZMASTER instance to the NTSGRP classes.

➢ **To connect the ZMASTER instance to the NTSGRP classes:**

1 From the **SYSTEMX** domain, choose the **ZPROCESS** class. The following window is displayed.



2 Choose the **ZMASTER** instance. The following window is displayed.



Scroll down in the attributes list box to an empty attribute field, and connect the groups by highlighting the empty attribute field and typing the connect Connect to To statement in the **Value:** text box. Then, select **Set Value**.

To connect to the first NTSGRP type:

```
SYSTEMX.NTSGRPS.&(ZNTSEC.ZNTGRP01)(EDMSETUP)
```

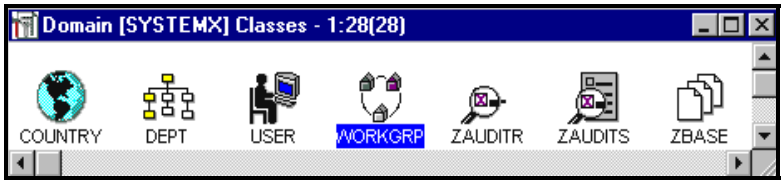
When you have completed all necessary connections, select **Update** to accept the changes you have made to the instance. To exit the instance without saving the changes, select **Cancel**.

Now that you have connected Windows NT security, you will want to use EDM to deploy applications. To do this, you will have to establish the workgroups in your Windows NT security schema to EDM. After you establish the workgroups, you can deploy applications to them by connecting their instances to ZSERVICE instances.

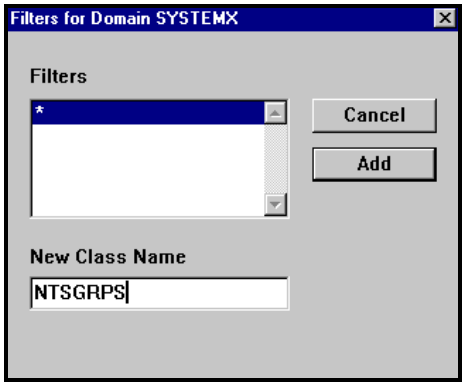
You need to know the names of all workgroups present in your Windows NT security schema. These names will be used when you manually create instances for the workgroups in the NTSGRPS class.

Establishing Windows NT Workgroups in EDM

1 From the **PRIMARY** file, choose the **SYSTEMX** domain.

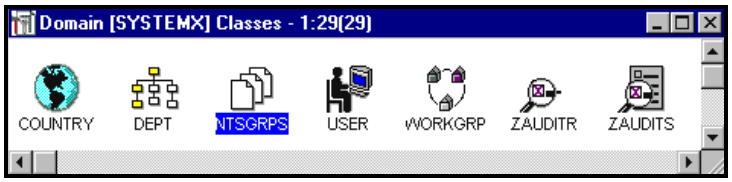


2 Select the **WORKGRP** class with your right mouse button, and choose **Copy Class** from the drop down menu that appears. The following screen dialog box is displayed.

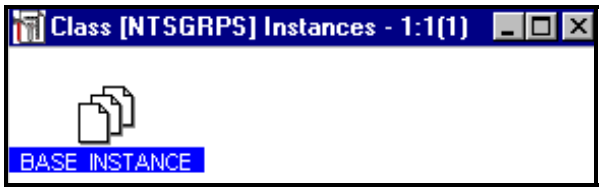


Type `NTSGRPS` in the **New Class Name** text box, and choose **Add**.

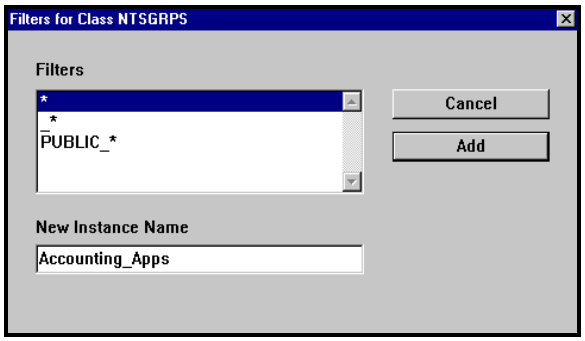
A class template window will appear. Choose **Add** from this screendialog box. The following window is displayed.



3 Choose the **NTSGRPS** class. The following window is displayed.

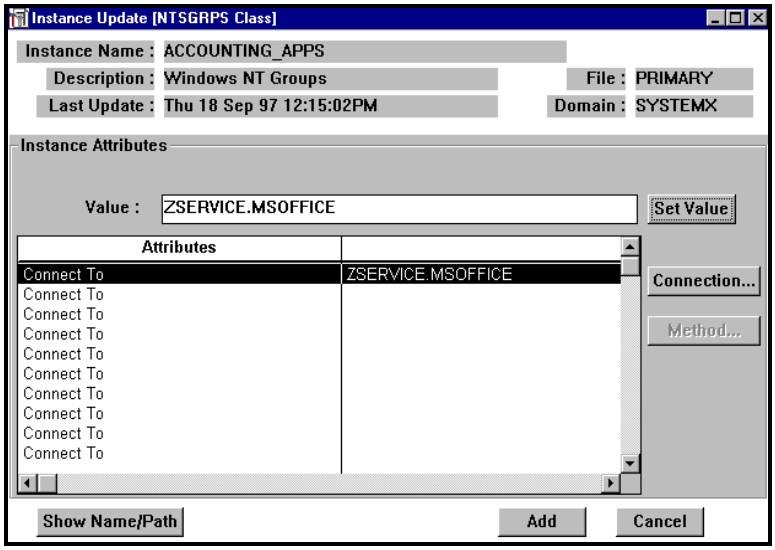


4 To add the instances for your Windows NT groups, select the BASE INSTANCE with your right mouse button, and choose **Copy Instance** from the drop down menu that appears. The following dialog box is displayed.



5 Type the name of the Windows NT group you want to add in the **New Instance Name** text box (i.e. `Accounting_Apps`), and choose **Add**. The following window is displayed.

Note: The name you use for the instance must identical to the name of the Windows NT group you are adding the instance for. If there is a space in the group's name, replace is with an underscore



6 Connect this instance to the ZSERVICE instances you want by selecting an empty **Connect To** attribute, typing the Connect To statement for the ZSERVICE you want in the **Value:** text box, and choosing **Set Value**.

A sample Connect To statement is shown below. This statement will connect an instance to the ZSERVICE instance for Microsoft Office the name of the instance in this database is MSOFFICE.

The Connect To statement in this case is:

ZSERVICE.MSOFFICE

You will need to perform steps **3** through **6** for each Windows NT group you want to control with EDM.

8 Using the EDM Console for Windows NT

This chapter provides you with directions for using the EDM Console facility for Windows NT. In particular, this chapter explains how EDM Console enables you to monitor and control the EDM Manager for Windows NT by monitoring system functions, issuing operational commands, and fine tuning system performance.

EDM Console at a Glance

The EDM Console for Windows NT gives you powerful monitoring and control capabilities for the EDM Manager, with an easy-to-use interface. EDM Console functions enable you to:

- Monitor system functions.
- View and adjust task settings and parameters.
- Submit REXX program requests.
- Notify (EDM Client) requests.
- Close down the EDM Manager.

Using EDM Console

The EDM Console for Windows NT operates from the **EDM Operator** folder on the NT platform. An EDM Console session is comparable to an EDM Client Connect process, and requires that communications be established and maintained between the EDM Manager and the EDM Administrator platforms.

> **Note**: The TIMEOUT_COMM setting of the MGR_TIMEOUT section of the EDMPROF.DAT file determines how long the EDM Manager will wait before disconnecting due to inactivity. Ensure that this value is set appropriately to maintain the EDM Console session without timing out.

You can use EDM Console to:

- Monitor complex system operations.
- Perform testing and benchmarking.
- Enhance system performance.
- Issue operational commands.

To access these functions, use the standard, familiar GUI components, such as menu bars and status buttons. Each EDM Console function provides distinct capabilities, as described in the following sections.

Navigating EDM Console

Navigating within the EDM Console is easy. The first screen displayed after opening the EDM Console is the **Selections** screen. The **Selections** screen lists the seven main Console functions.

To choose a function from the **Selections** screen, highlight the function and press ENTER.

To return to the **Selections** screen from a Console function screen, simply select **Close**, or press the **Selection** screen button on the status bar.

> **Warning**: For the purposes of the EDM Manager Version 4.0, do not try to perform more than one Console task at the same time. This will result in Console termination.

Accessing EDM Console

Since EDM Console provides powerful tools that directly effect system operations, EDM Console is not automatically accessible to all EDM users. You can determine who has access to some or all EDM Console functions by establishing individual access levels.

There are two ways of establishing access to EDM Console; by configuring a user to appropriate access levels, or determining access through the EDM Manager‘s EDMPROF.DAT file.

> **Note**: For the purposes of the EDM Manager Version 4.0, please determine access using the EDMPROF.DAT file. Instructions for configuring user access will be provided with subsequent versions of this documentation.
>
> To provide access to all EDM Console functions, change the CONSOLE setting of the MGR_ACCESS section to IGNORE. Then, restart the EDM Manager to make this setting take effect.
>
> For more information about the EDMPROF.DAT file, see “Chapter 3, *: "Tuning the NT EDM Manager*.""

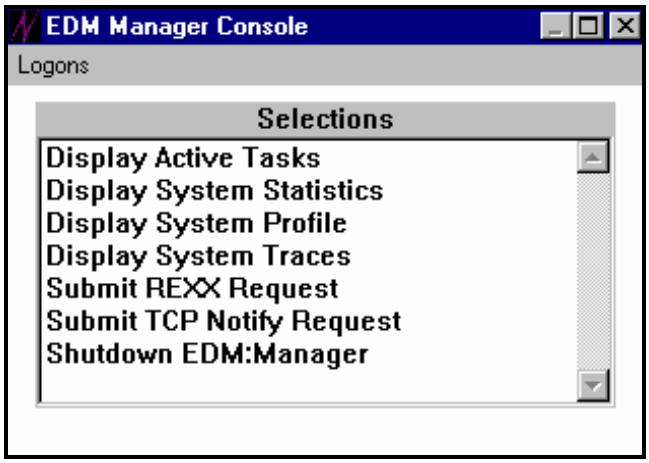
Opening EDM Console

➢ **To Open EDM Console:**

1 From the **Start** menu select **Programs**, **EDM Operator**, and **EDM Console**.

2 The EDM Console Security Information dialog box is displayed. Enter your User ID and Password, then click **OK**

The **Selections** window screen is then displayed in the **EDM Manager Console** window..



The **Selections** screen displays the seven main Console functions as described in the table below.

EDM Console Functions

Function Name	Description
DISPLAY ACTIVE TASKS	Lists active tasks, and enables you to detach a specific task, and view or change the trace flag status for a particular task.
DISPLAY SYSTEM STATISTICS	Displays the current system status as defined by various system processes.
DISPLAY SYSTEM PROFILE	Displays EDM Manager settings from the EDMPROF.DAT file.
DISPLAY SYSTEM TRACES	Displays the current system trace settings.
SUBMIT REXX REQUEST	Enables you to execute a REXX program.
SUBMIT TCP NOTIFY REQUEST	Allows you to notify an EDM Client to perform an EDM Client Connect.
SHUTDOWN EDM Manager	Enables you to shut down the EDM Manager.

Note: Two additional functions, Trace Settings and Select Logon Types, are accessible only through main Console functions: to get to Trace Settings, use Display Active Tasks, and to get to the Select Logon Types, use the Logons menu of the Selections screen.

➢ **To Choose an EDM Console Function:**

1 Using the left mouse button, double click the function. The screen for the function you have chosen is displayed.

2 To exit the EDM Console facility, choose **Exit** from the **Logons** pull down menu of the **Selections** screen, or select the Close button in the upper right corner of the screen.

The following sections describe the Console functions.

Select Logon Types

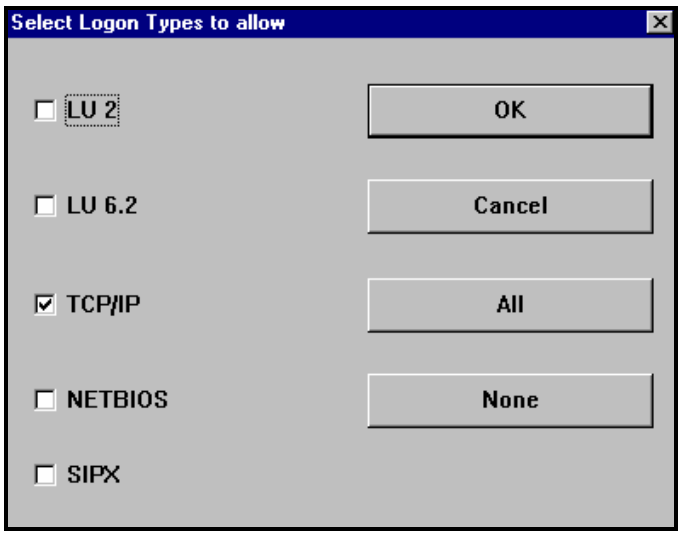
The Select Logon Types option allows you to determine which EDM Clients will be notified to perform an EDM Client Connect process. The selection is based on the EDM Client protocol type. This function allows you to select one or all of EDM Manager-supported communications types.

The Select Logon Types option enables you to choose which specific EDM Clients, based on supported communications, you allow to connect. You can use this function to monitor system performance and troubleshoot by permitting certain EDM Clients to connect, and measure the execution. In addition, you can resolve communications problems, or enhance system efficiency, by preventing types of Clients from connecting.

➢ **To Access Select Logon Types:**

1 Choose **Allow...** from the Logons pull down menu on the **EDM Manager Console** Selections window.

2 The following dialog box is displayed. It shows the protocols, **LU2, LU6.2, TCP/IP, NETBIOS**, and **SIPX**.



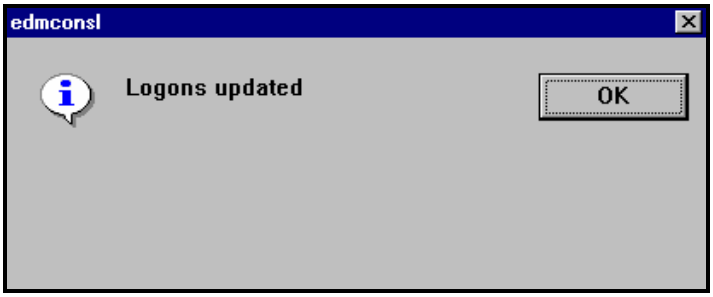
3 You may:

- Select all protocol types by clicking on **All.**
- Select none of the protocol types by clicking on **None**. Note that selecting **None** will prevent EDM Client Connects from taking place.
- Select protocol types one at a time by clicking on any or all of the boxes on the left. A check mark will appear.

4 When your selections have been made click **OK** to save them.

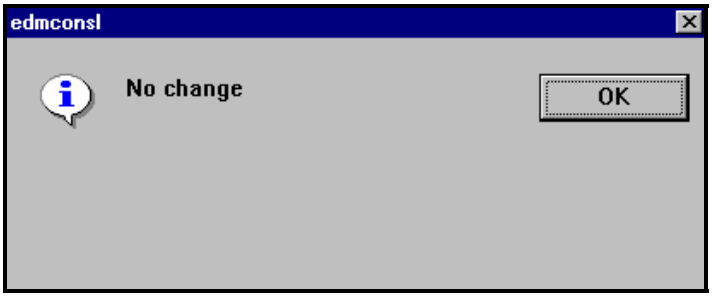
Note: If you have selected **None**, and try to exit the Console session, you will be reminded that you have disabled all logons. You should restore some logon types before exiting the Console session.

- If you have made changes, the following dialog box appears.:



Confirm that you have made changes by clicking **OK**.

- If you have not made changes, the following dialog box appears:.



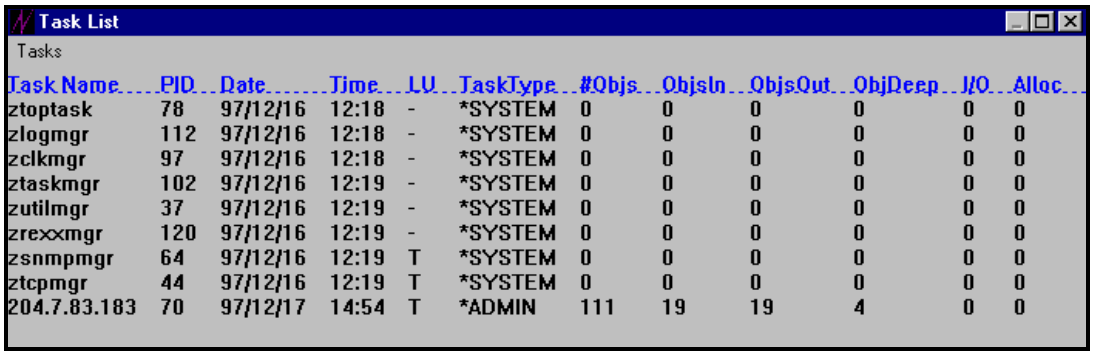
Confirm by clicking **OK**.

5 If you do not want to save any changes you have made, click **Cancel.** The **Select Logon Types to allow** dialog box will be closed and you will return to the **EDM Manager Console** window.

Display Active Tasks

Display Active Tasks shows a list of currently active tasks, with details about each one. It allows you to see who is logged on, what processing is being done, and what system processing anomalies are present. You may also remove (detach) a task from the list of active tasks.

When you select **Display Active Tasks** from the **EDM Manager Console** window theThe **Display Active TasksTask List** window is displayed., which is shown below, It shows tasks in rows across the screen. The Z-named tasks are the EDM Manager system tasks that are currently running. EDM Client tasks are denoted by their IP addresses.



Task Name	PID	Date	Time	LU	TaskType	#Objs	ObjsIn	ObjsOut	ObjDeep	I/O	Alloc
ztoptask	78	97/12/16	12:18	-	*SYSTEM	0	0	0	0	0	0
zlogmgr	112	97/12/16	12:18	-	*SYSTEM	0	0	0	0	0	0
zclkmgr	97	97/12/16	12:18	-	*SYSTEM	0	0	0	0	0	0
ztaskmgr	102	97/12/16	12:19	-	*SYSTEM	0	0	0	0	0	0
zutilmgr	37	97/12/16	12:19	-	*SYSTEM	0	0	0	0	0	0
zrexxmgr	120	97/12/16	12:19	-	*SYSTEM	0	0	0	0	0	0
zsnmpmgr	64	97/12/16	12:19	T	*SYSTEM	0	0	0	0	0	0
ztcpmgr	44	97/12/16	12:19	T	*SYSTEM	0	0	0	0	0	0
204.7.83.183	70	97/12/17	14:54	T	*ADMIN	111	19	19	4	0	0

The entry for each task contains pertinent data about that task. These parameters are described in the table below.:

Active Task Parameters

Parameter	Description
Task Name	Name of task from task list.
Process ID	Task ID.
Date and Time	Date and time task was initiated.
Logical Unit	Protocol type (TCP/IP, Net Bios, SIPX., LU 2, LU 6.2
Task Type	System, Client, or Admin.
#Objs	Number of objects associated and used with that task.
ObjsIn	Number of objects received from the Manager.
ObjsOut	Number of objects delivered to the Manager.
ObDeep	Maximum depth of resolution.
I/O	Number of bytes pulled out of the database.
Allocation	Number of resource files sent to the Client.

You can use Display Active Tasks in a number of ways. For example, it can show you the progress and duration of EDM Client Connects. This function can alert you to the status of system tasks that might impact EDM Manager performance. You can also stop an EDM Client Connect in progress in order to reallocate system processing resources or fine tune overall performance.

The **Tasks** menu of the **Task List** screen window has the following four selections:

- **Refresh**

  Updates the Display Active Tasks information for all displayed tasks.

- **Trace Settings**

Allows you to view and change the EDM Manager log settings associated with that task. You must first select a task from the list to use this option. For more details, see "*Trace Settings*." on the next page.

- **Detach**

  Enables you to stop a task that is currently running. You must first select a task from the task list to use this option. Note that you are not able to detach a system task, as this would affect subsequent EDM Manager operations. You can only stop EDM Client tasks.

- **Close**

  Closes the **Display Active TasksTask List** window and returns you to the **Selections screenEDM Manager Console** window.

Trace Settings

Trace Settings displays the current trace settings for the specified task. The trace settings are used to control the amount of information that is sent to the EDM Manager log about this task. The Trace Settings function, together with the EDM Manager log, enables you to monitor system performance. You can use this option to increase the volume of information that is generated about critical processes, and decrease reporting on lower priority tasks.

You can scroll down the screen to see all of the trace settings. Not all settings have been defined or are applicable to the NT EDM Manager. For more information about trace settings and the EDMPROF.DAT file, see "Chapter 3, :" *Tuning the NT EDM Manager*."

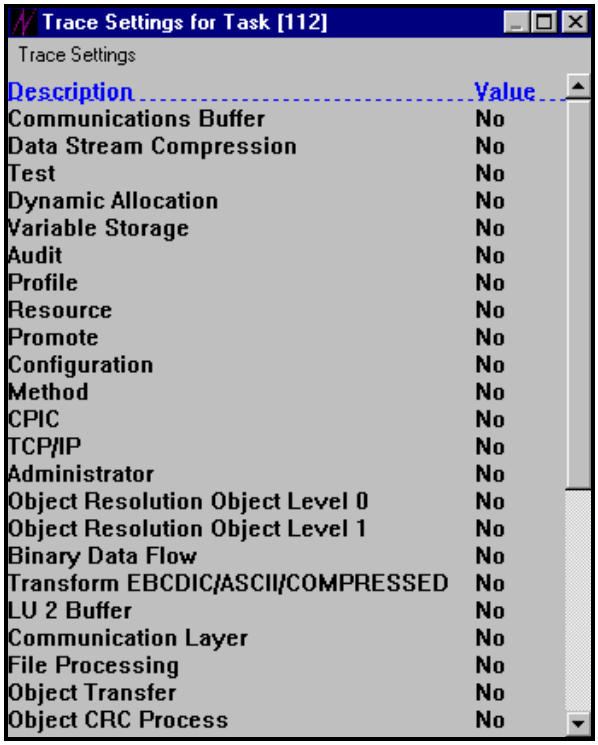
Each setting is displayed as **YES Yes** if the trace setting is turned on, and **NO No** if the setting is turned off.

➢ **To Access Trace Settings:**

1 You must first choose a task from the **Task List** screen. To do this, highlight the selection by clicking on it.

2 Open the **Tasks** menu from the **Task List** screenwindow.

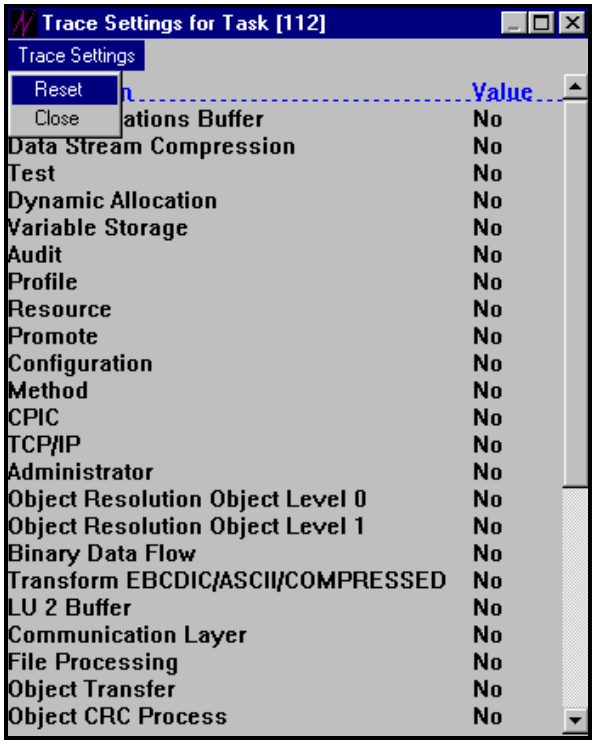
Click on **Trace Settings**. The following window is displayed.



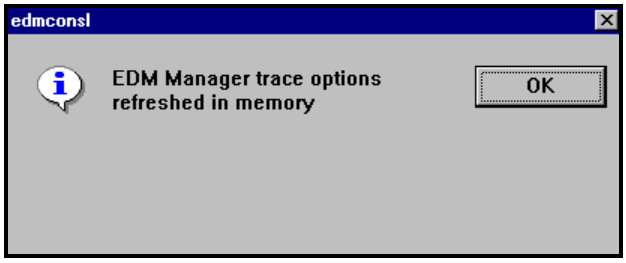
➢ **To Change the Trace Setting Value of an Item in the Description List:**

1 Double click on the item whose value you want to change. You will be able to toggle back and forth between **Yes** and **No**.

2 After all desired changes have been made, click on **Reset** from the **Trace Settings** pull down menu to set the values you have chosen.

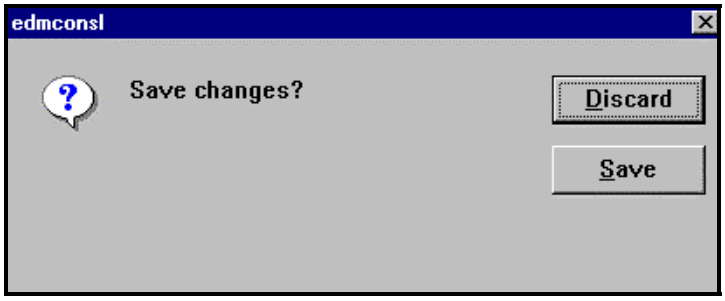


The following dialog box is displayed:.



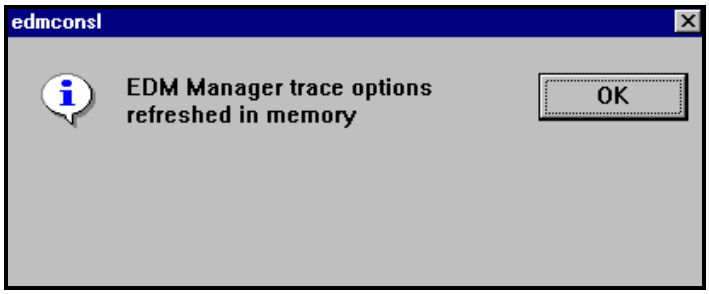
3 Then select **Close** from the same pull down menu.

The following dialog box is displayed.:



4 Choose **Discard** to reject the trace options.

5 Choose **Save** to retain the trace options.

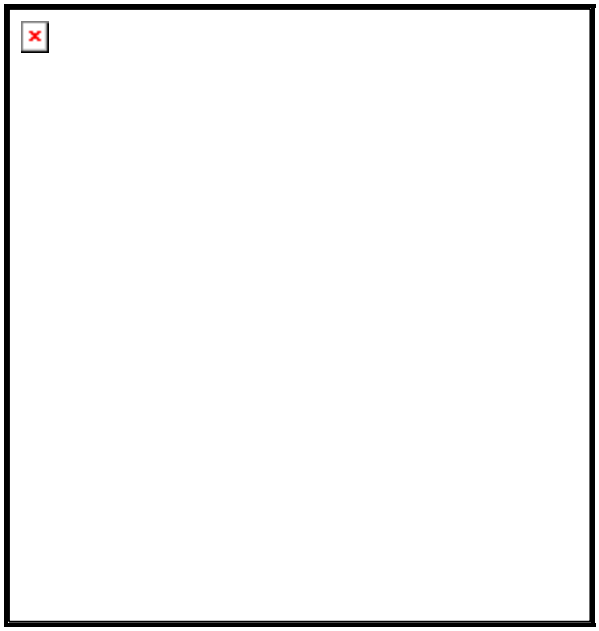


The **Trace Settings** menu has the following two selections:

- **Task**
  Allows you to individually change any or all of the trace settings for a particular task. To change a trace setting, place the mouse pointer on the task; you do not have to highlight the line. Then, double-click the mouse to toggle between **Yes** and **No**.
- **Global Default**
  Allows you to change all of the trace settings for a particular task to those found in the EDM Manager‘s settings in the MGR_LOG section of the EDMPROF.DAT file.

Display System Statistics

Display System Statistics allows you to view the current status of the EDM Manager. The information listed here is the samesimilar to the data shown on in the **Display Active TasksTask List screen** window, but summed for all tasks, with some additional system status details. Note that data are for completed processes only. On-going tasks will not be represented. Use **Display Active Tasks** to view data for on-going processes.



You can use **Display System Statistics** to monitor both overall system performance and EDM Manager performance. This function also enables you to collect detailed statistics for testing and benchmarking.

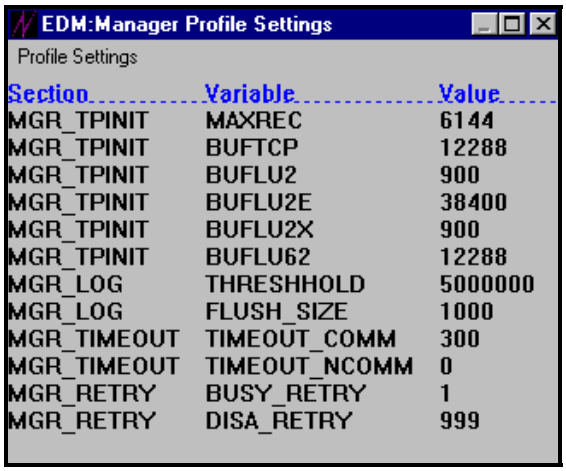
The **Statistics** menu has the following two selections:.

- **Refresh**
- Allows you to update the display.
- **Close**
- Allows you to exit the Display SSystem Statistics window.

Display System Profile

Display System Profile displays a list of the current settings (Variables) in the EDMPROF.DAT file's MGR_TPINIT, MGR_LOG, MGR_TIMEOUT, and MGR_RETRY sections. Specifically, the displayed information provides the:

- Buffer size for all supported communications protocol types.
- Values that determine the retry parameters for EDM Manager-EDM Client sessions.
- Size and flexibility of your EDM Manager log.



You can use **Display System Profile** to validate system settings when performing troubleshooting or monitoring system performance. For example, the settings for both the MGR_RETRY and MGR_TIMEOUT sections may point out problems with failed EDM Client Connect attempts. Likewise, MGR_TPINIT settings can indicate communications difficulties. Also, you can use the MGR_LOG settings to determine if your EDM Manager log is adequately configured to provide appropriate feedback information for your EDM operations.

The **Profile Settings** menu has the following two selections:

- **Reset**
- This option is not currently available.
- **Close**
- Allows you to exit the Display System Profile window.

Display System Traces

Display System Traces shows the current trace settings for the EDM Manager log as specified in the MGR_TRACE section of the EDMPROF.DAT file. The trace settings are used to control the volume and type of information that is sent to the EDM Manager log for all log parameters.

The **Trace Settings** option, together with the EDM Manager log, enables you to monitor system performance. You can use this option to increase the volume of information that is generated about critical processes.

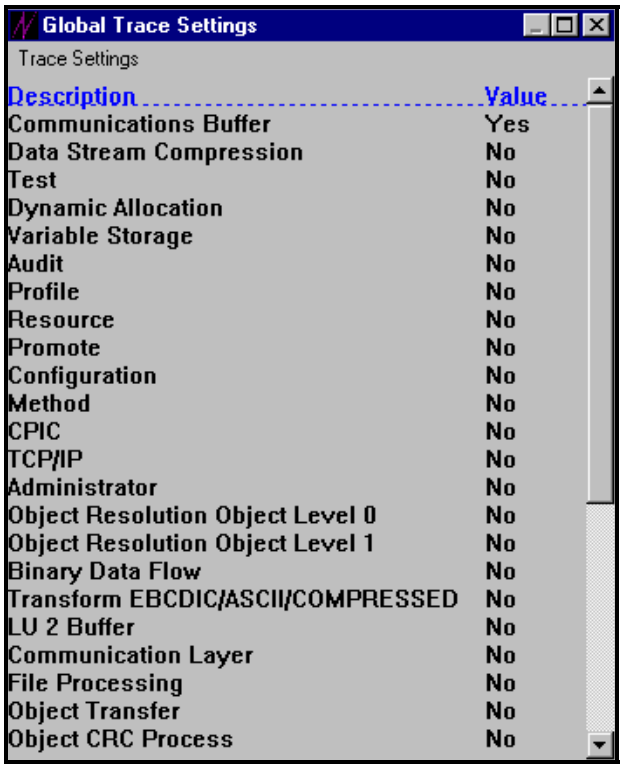
Each setting is displayed as **Yes** if the trace setting is turned on, and **No** if the setting is turned off.

To see all of the trace settings, scroll down the screen. Not all settings have been defined or are applicable to the EDM Manager for Windows NT. For more information about trace settings and the EDMPROF.DAT file, see "Chapter 3*: "Tuning the EDM Manager for Windows NT*."

➢ **To Open the Display System Traces Dialog Box:**

3 Click on **Display System Traces** inon the **Selections** screen of the **EDM Manager Console** window.

4 The **Global Trace Settings** window is displayed.

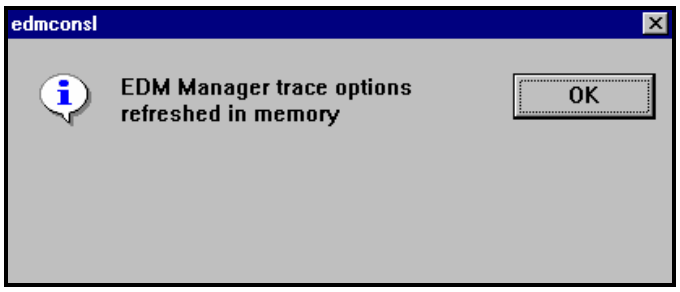


➢ **To Change a Trace Setting:**

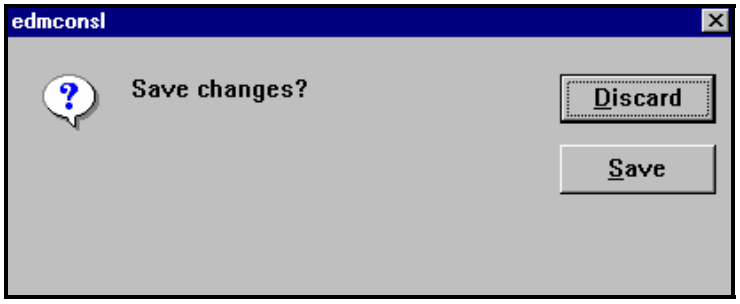
1 Place the mouse pointer on the desired task; you do not have to highlight the line. Double click the left mouse button to toggle between **Yes** and **No**.

2 After all desired changes have been made, select **Reset** from the **Trace Settings** pull down menu to set the new values you have chosen.

The following message is displayed:.

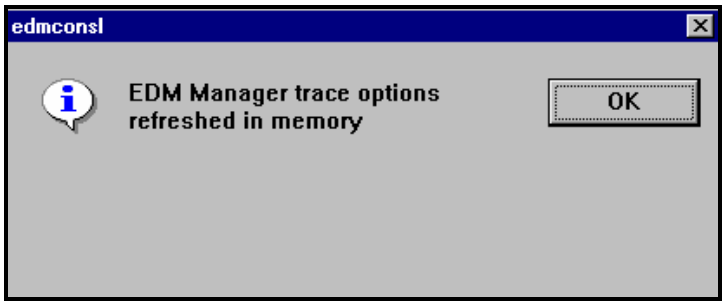


3 Select **Close** from the **Trace Settings** pull down menu to close the **Global Trace Settings** dialog boxwindow. If you have made any changes but have not saved them using **Reset**, the following message is displayed:.



4 Choose **Discard** to reject the trace options, or choose **Save** to retain the trace options.

5 If you have saved your changes, the following message is displayed.:

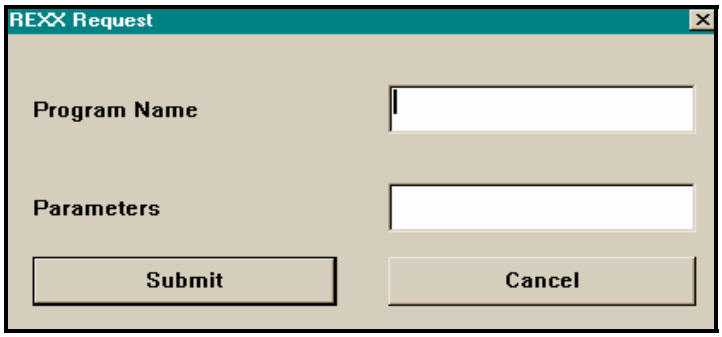


6 Click **OK** to confirm. You will be returned to the **Task List** window.

Submit REXX Request

Submit REXX Request allows you to initiate REXX programs. You can specify the REXX program you want to execute, as well as any permissible parameters. To identify where the EDM REXX directory is, consult the REXX_PATH value in the MGR_DIRECTORIES section of the EDMPROF.DAT file.

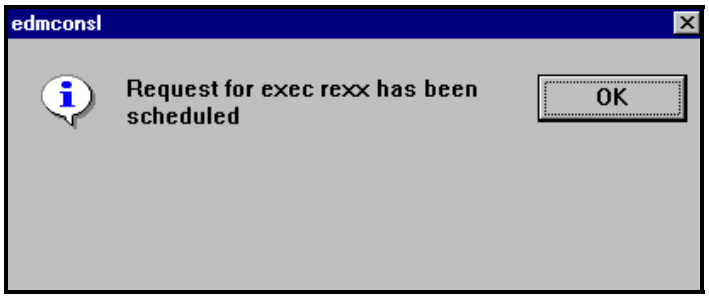
> **Note**: **The Program Name** field must contain a REXX program that is contained in the EDM REXX directory.



Submit REXX Requests enables you to customize EDM Manager functioning. You can insert a REXX program at given points in EDM Manager operations to achieve specific results. For a full listing of the EDM provided Manager Methods, refer seeto Appendix A:, "*The EDM Manager Operations Guide for Windows NT, Appendix A, EDM Manager Methods Reference.*"

While the Submit REXX Request function has no menu options, there are two function buttons.

Choose **Submit** to initiate the REXX program. The following message is displayed:.



Choose **Cancel** to clear the Submit REXX Request entry.

Submit TCP Notify Request

The **Submit TCP Notify Request** function allows you to direct an EDM Client to connect to the EDM Manager. This permits you to perform on-demand resolution and configuration of EDM Clients.

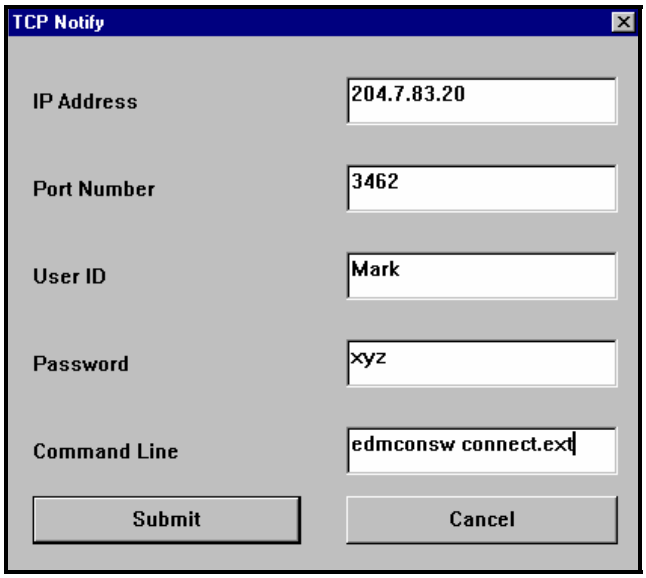
The Submit TCP Notify Request function enables you to perform individual, on-demand, EDM Client resolution for use in monitoring system performance or benchmarking. Also, by specifying specific command lines, you can further customize the EDM Client Connect process.

To initiate the EDM Client Connect, you must supply the following parameters:

Parameters

Parameter Name	Description
IP Address	The EDM Client IP address.
Port Number	The TCP/IP socket that the EDM Client is using. This value should be 512 unless otherwise specified as the ZNTFPORT value for that EDM Client. .
User ID	The ZUSERID in the EDM Client ZMASTER object.
Password	The EDM Client password as specified in the ZMASTER object.
Command Line	Can be any executable program residing on the EDM Client desktop (REXX or EXE).

Note: All five Submit TCP Notify Request parameters must be specified in order for this function to operate.

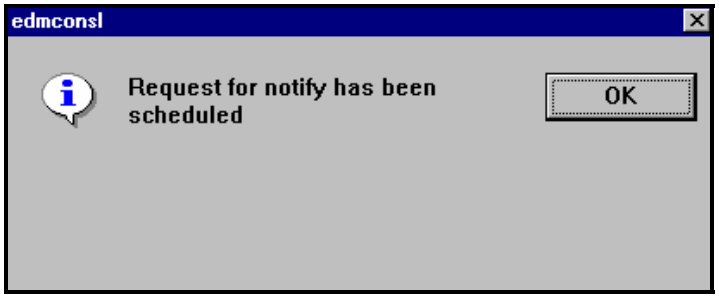


➢ **To Submit TCP Notify Request:**

1 Fill in the five parameters described above.

2 Then click **Submit** to execute your request.

The following message is displayed.



Click **OK** to confirm.

3 Or click **Cancel** to in the **TCP Notify** dialog box to cancel and return to the **EDM Manager Console** window**.**

The **TCP Notify** menu has one selection:

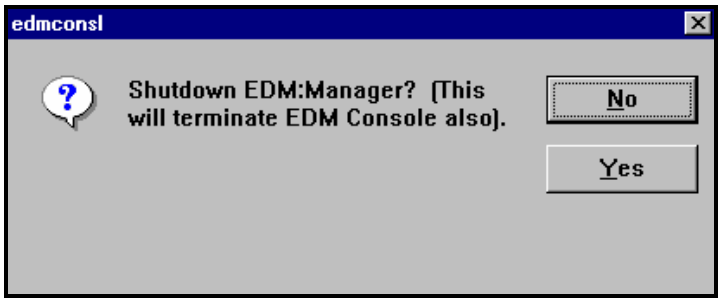
- **Select Logons**
-

  Brings up the **Select Logon Types** window.

Shutdown EDM Manager

The **Shutdown EDM Manager** function provides an alternate means of ending EDM Manager processing. For more information on stopping the EDM Manager, see “Chapter 2: , " *Starting and Stopping the EDM Manager for Windows NT*.”"

The **Shutdown EDM Manager** window has the following two buttons.:



Choose **No** to cancel the shutdown request.

Choose **Yes** to clear the selectionshutdown the EDM Manager.

9 The EDM Performance Monitor for Windows NT

This chapter explains how to use the EDM Performance Monitor. It provides you with the ability to obtain information about a variety of EDM Manager related tasks that can help you assess the performance of your EDM Manager.

Overview

The EDM Performance Monitor allows you to obtain information about the performance of your EDM Manager. It makes it possible to determine the efficiency of the EDM Manager and its optimal operational configuration.

You can also use this tool for problem determination. For example, the Performance Monitor can be used to observe the compression ratio of data in the database. By tracking the effects of different compression ratios, you can determine the ratio that generates the most efficient database access time.

The EDM Performance Monitor enables you to compare the rate at which it is processing various EDM Manager tasks. This tool will provide you with information on the rate at which the EDM Manager is processing operational maintenance tasks (Tasks), the communications performance of the EDM Manager through protocols such as TCP/IP (TP), and the efficiency with which the EDM Manager is accessing information from the database.

The EDM Performance Monitor can also be used to observe other operations, such as the object resolution process, data compression and decompression, and method processing.

Opening the EDM Performance Monitor

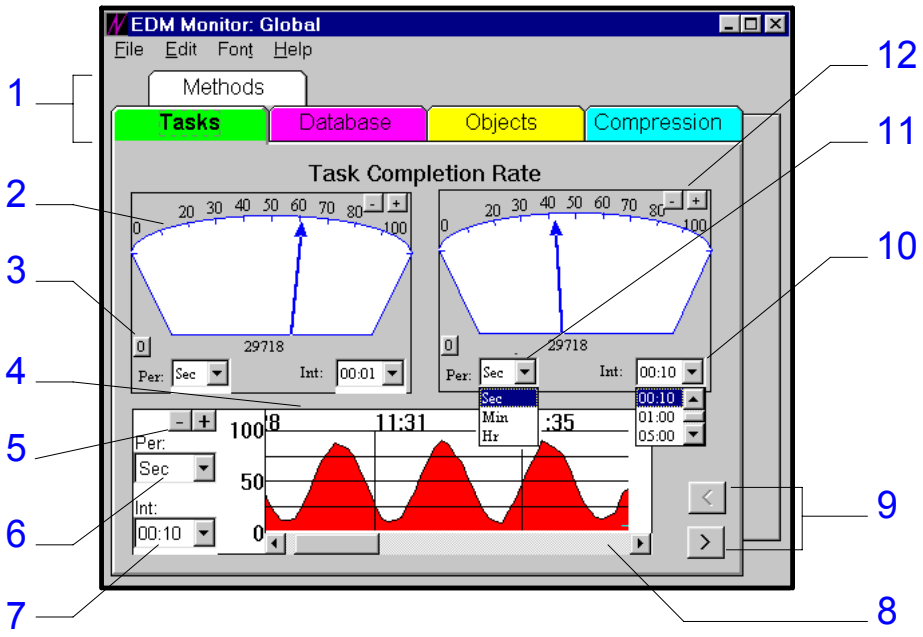
The EDM Performance Monitor has a graphical display that will enable you to quickly and easily view different performance aspects of your EDM Manager. This section will explain the EDM Performance Monitor interface.

➢ **To Access the EDM Performance Monitor:**

- From the **Start** menu, select **Programs**, **EDM Operator**, **EDM Monitor**.

The EDM Performance Monitor Display

The figure below has been broken down into its different components. A brief description of each component is provided in the table below.



The main menu of the Performance Monitor provides you will several options. This menu will be useful when using the EDM Performance Monitor.

You can use the main menu to exit the performance monitor. To exit the monitor, select **File** from the menu, then select **Exit** from the drop drop down menu that appears.

The font used in the Performance Monitor displays can be changed by selecting **Font** from the main menu. Use the dialog box that appears to change the font characteristics of your display.

EDM Performance Monitor Display Areas

Area	Description
1	Each tab in this area of the display represents an aspect of the EDM Manager that the EDM Performance Monitor can use to observe data. For more information about tabs, see the section "*TabAB Selections*."
2	The numbers on this display show the units of time for the data output. The range for this display is defined in area 12.

Area	Description
3	This option is used to reset the time for this report field.
4	A time stamp is displayed on the chart at this location for each complete time interval. The time interval is defined in areas 6 and 7.
5	The - and + options can be selected to change the numerical range of the chart.
6	This drop down list box is used to change the units you are using to measure the time interval for data collection. In some views, this list box is used to select the value of a statistical report. A report can be made on the average, maximum, or minimum value.
7	This drop down list box can be used to change the time interval for report markers in the chart. in the chart.
8	This scroll bar can be used to scroll through the graphical chart that is generated from the information of the EDM Performance Monitor.
9	These option buttons can be used to switch the view of your data to other views for the TAB tab selections. For example, in the **Tasks** TABtab, these options buttons are used to switch from monitoring the **Task Completion Rate**, to monitoring the **Number of Active Tasks**.
10	This drop down list box can be used to select the time interval value you will use to periodically summarize the data you are collecting. The intervals available are 00:10, 01:00, and 05:00. For example, if you have chosen **Sec** for your Period, 00:10 will represent an interval of ten seconds.
11	You can use this drop down list box to select the units you are using to measure your time interval. The units available are Seconds, Minutes, and Hours. In display views in which you are viewing statistics over a given time period, this will change from **Int** to **Stat**. The drop down list box is used to view statistics according to average, maximum or minimum values in a given time period.
12	You can select the - and + options to increase or decrease the numerical range for the graphical display of your data.

Tab Selections

Each tab in the main display of the EDM Performance Monitor provides you with the ability to monitor a different operational aspect of the EDM Manager.

Tasks

The **Tasks** Tab tab is used to monitor the performance of tasks that are needed to run the EDM Manager. For example, when an EDM Client runs the EDM Client Connect Process, it is considered to be a task.

There are two different ways to observe the Tasks your EDM Manager is processing.

- **Task Completion Rate**.

  This display allows you to monitor the rate at which your EDM Manager is processing tasks.

- **Number of Active Tasks**.

  This display allows you to observe the number of active tasks currently running on your EDM Manager.

Database

The **Database** tab is used to monitor how efficiently your EDM Manager is accessing information from the database.

The **Database** tab provides many different views:

- **Database Bytes Read Rate**.

This is used to monitor the rate at which your EDM Manager is reading from the database.

Database Get Operations Rate.

This is used to observe the rate at which your EDM Manager is retrieving data from the database.

- **Database Update Operations Rate**.

  This view provides you with information about the rate at which operations are being updated.

- **Database # Files Read Rate**.

  This view can be used to find the number of files that are being read from the database in a given period of time.

Objects

This The **Objects** tab is used to monitor the object resolution process on your EDM Manager in three different formats.

- **Object Resolution Rate**.

  This view allows you to observe the rate of object resolution on your EDM Manager.

- **Object In Rate**.

  This view is used to monitor how fast your EDM Manager is processing new objects.

- **Object Out Rate**.

  This view gives you the ability to observe how fast your EDM Manager is processing outgoing objects.

Compression

This The **Compression** tab lets you monitor data compression and data decompression on your EDM Manager in several ways.

- **Compression Operations Rate**.

  This selection gives you the ability to monitor the rate of data compression.

- **Compression Ratio State**.

  This selection allows you to monitor the compression ratio for data.

- **Decompression Operations Rate**.

  This selection gives you the ability to monitor the rate of data decompression.

- **Decompression Ratio State**.

  This selection allows you to monitor the compression ratio for data.

Methods

This The **Methods** tab is used to monitor the rate and number of methods the EDM Manager is executing. It may be useful to monitor this to determine the efficiency of your EDM Manager.

You can use this tab to monitor the REXX Methods rate, or the Compiled Methods rate.

- **REXX Methods Rate**.

  This view will display the rate REXX methods are processed by your EDM Manager.

- **Compiled Methods Rate**.

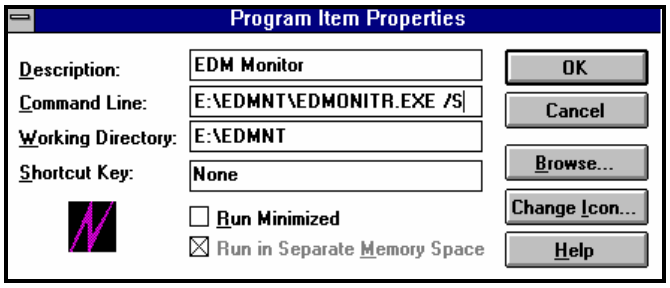
  This view will display the rate at which methods are compiled by your EDM Manager.

Editing the Target Executable

There are several options available to you when opening the EDM Performance Monitor. These options are contained within the executable statement, and provide you with the ability to control your monitor display view, and data collection source - EDM Manager or an EDM Client.

If you want to use an executable option when opening the EDM Performance Monitor through the icon or Windows Taskbar, you must edit the target executable statement. To edit the target executable statement:

1. From the **EDM Operator** program group, highlight the **EDM Monitor** icon.
2. Choose **File** from the main menu bar.
3. Choose **Properties** from the **File** submenu. The following dialog box is displayed.

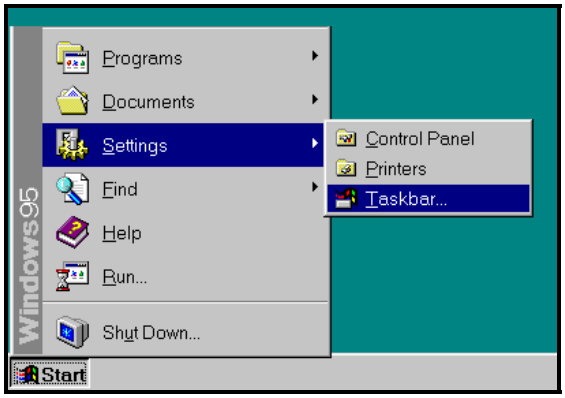


4. Type the executable option at the end of the statement in the **Command Line** text box and choose **OK**.
5. Choose **EDM Monitor**.

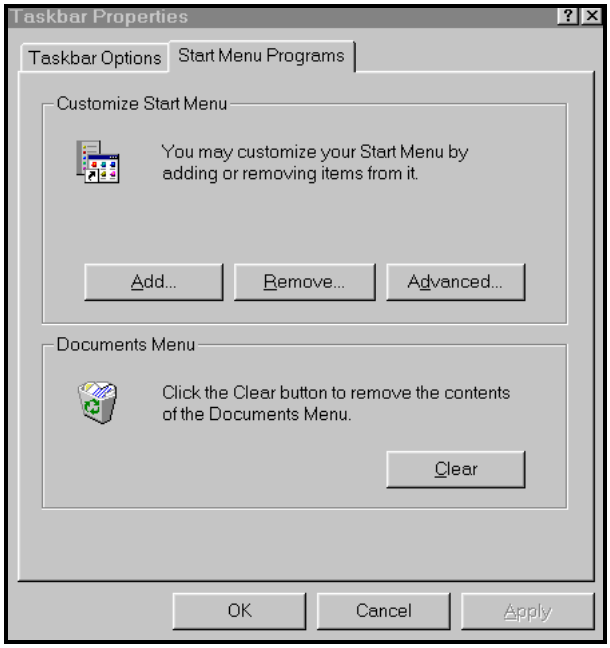
Editing the Target Executable

➢ **To Edit the Target Executable Statement:**

1. From the **Start** menu, select **Settings**.

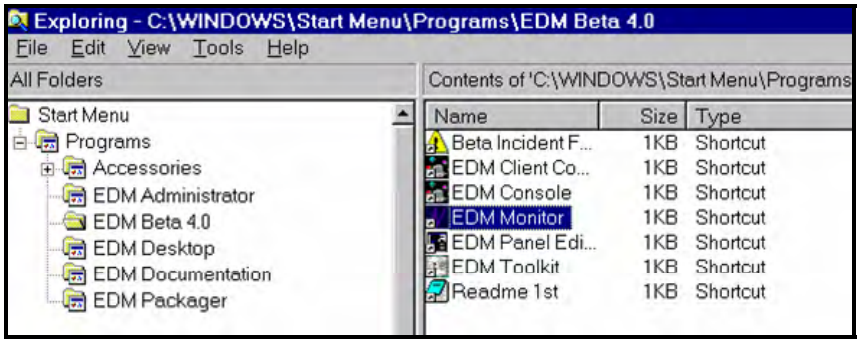


2 From the **Settings** submenu, select **Taskbar**. The following dialog box appears.



3 Select the Start Menu Programs tab from the **Taskbar Properties** dialog box.

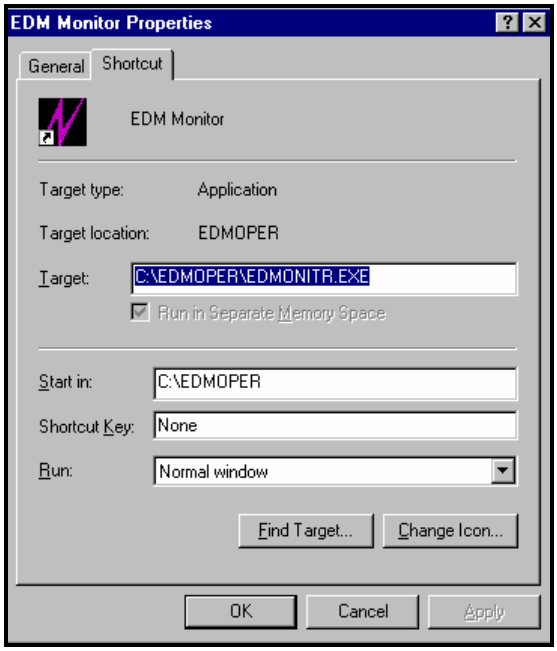
4 Choose the Advanced option. The following screen appears.



Choose **Programs** from the tree view, then choose the **EDM Operator** folder.

5 Select **EDM Monitor** with your right mouse button, and choose **Properties** from the drop down menu that appears.

6 Select the **Shortcut** tab from the **EDM Monitor Properties** dialog box.



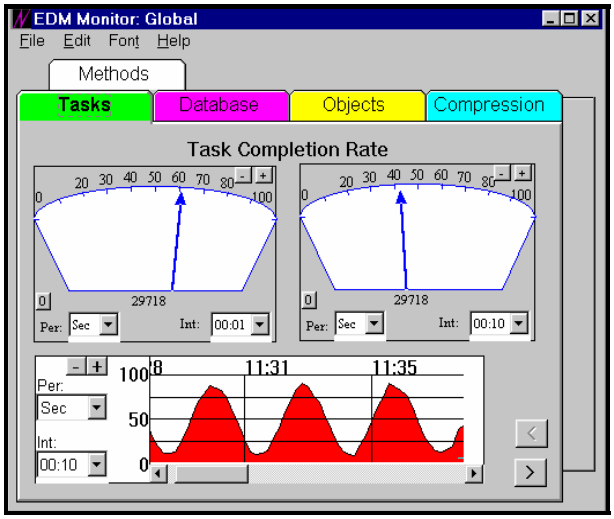
7 Type the executable option at the end of the statement in the **Target** text box and choose **OK**.

Executable Options

Option	Definition	Description
/I + a number	Interval	This option controls the data interval that is communicated to the EDM Manager. The default value for this is 1, with a maximum value of 99.
/m	Monochromatic Tabs	Normally, the tabs on the monitor display are different colors. This option will display the tabs in a monochromatic fashion.
/on	On demand	When you specify this option, you are telling the EDM Performance Monitor to collect data for the initial display screen. Other data will be collected only when you access the information from its tab on the main menu.

EDM Performance Monitor Preferences

When you enter the proper executable statement from the command line, you will be presented with the main EDM Performance Monitor display. The EDM Performance Monitor display window is shown below.



While using the EDM Performance Monitor, you may want to change the display of its data.

The Performance Monitor window is highly flexible, and will allow you to adjust, or detach, the windows of its display.

All windows in the main display are detachable, and can be removed in one of two ways. To remove a window:

- Place the mouse pointer on the window you want to move, and use the left button to drag the window to its new location.
- Select the window with your right mouse button. Select **move Move to window** from the drop down list box that appears. The widow will appear at the lower left corner of your screen.

All detached windows are fully functional, and maintain usability even after you have switched to another tab view.

You may want to maximize the graphical view form of a window. This is especially desirable if you have made a particular window extremely small. You can hide the buttons from a window by selecting it with the right mouse button, and choosing **hide Hide controls** from the drop down list box that appears.

To reattach a window to the main display, select the window with the right mouse button and choose **move Move to window** from the drop- down menu.

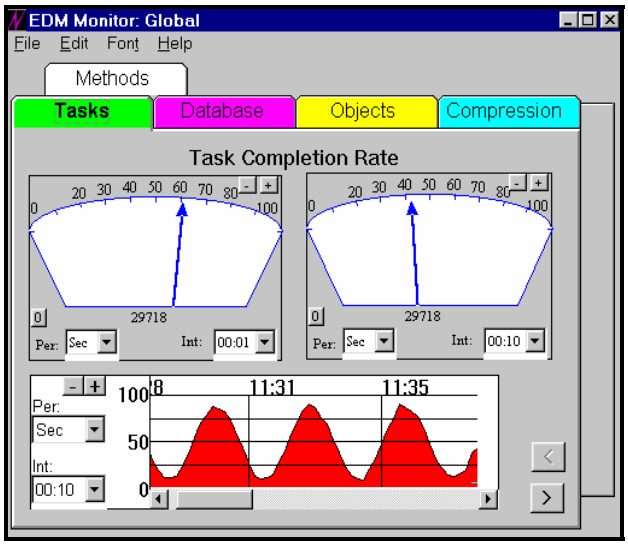
Opening the EDM Performance Monitor from a Command Line

➢ **To Access the EDM Performance Monitor:**

1. Select **File** from the main menu in the Program Manager.
2. Select **Run** from the drop down menu that appears.
3. In the **Run** text box, type

   `edmonitr [/option].`

   The EDM Performance Monitor is displayed:.



Note: In Windows NT 4.0, select **Run** from the **Start** menu, and type the executable statement in the dialog box that is displayed.

Example

```
edmonitor /s
```

Note: You will also need to type the path leading to the edmonitr executable file. The path is:

C:\EDMOPER\

The **Browse** option can also be used to locate the executable file.

Example

```
c:\edmoper\edmonitor /s
```

In the examples above, the EDM Performance Monitor will obtain data from a simulated EDM Manager session. The data used in the simulation is gathered from the EDM Client the monitor is being executed from.

More than one option can be called from the initial command line.

Note: When you specify several options, you must include the backslash with each. You must also leave a space between each option.

Example

```
edmonitor /s /m.
```

In the example above, the EDM Performance Monitor will obtain data from a simulated EDM Manager session, and will display all tabs in a monochromatic fashion.

10 Manager Add-On Features

This chapter contains information about additional features that can be used with the EDM Manager. You can find information about features such as SMTP mail support and client error message retrieval.

EDMMAILQ

EDMMAILQ is a manager method that deposits e-mail in the Mail Queue (OutBox). Once deposited, e-mail can be, retrieved by the SMTP Send Manager (zsmtsmgr) and sent to a user on a remote system.

If the zsmtsmgr task is not included in the [MGR_ATTACH_LIST] section of the EDMPROF.DAT file, e-mail will not be delivered to the remote user.

A list of parameters is provided in the table below.

Parameter Name	Description
FROM	Required. The sender's address (i.e. user1@company1.com).
MESSAGE	Required. Specifies a brief message (limited to 255 characters).
TO	Required. Specifies the Ee-mail recipients. Multiple users can be listed by using ';' as the delimiter between each recipient. (i.e. user1@company1.com;user2@companyb.com).
ATTACH	Optional. To sSpecifies attachment files. Multiple attachments can be listed by using ';' as the delimiter between each attachment (i.e. c:\config.sys;c:\autoexec.bat). Attachments are sent using MIME. Attachments can be in binary format.
MESGFILE	Optional. Specifies the file that contains the message. Is used in place of the MESSAGE parameter if the message is greater than 255 characters.
SUBJECT	Optional. Specifies the subject of the Ee-mail.

Note: Arguments are expected to be **KEYWORD=VALUE** format delimited by commas.

Parameters are used as Keywords and are not case sensitive.

Usage

In the example below mail is sent from **user1@company1.com** to **user2@company.com** with a brief message.

Example

```
EDMMAILQ from=user1@company.com,to=user2@company2.com,Message=
"This is a brief message"
```

Note: The double quote character is needed when a value contains embedded blanks, or commas.

In the example below, a text file (c:\report.txt) is sent from **user1** to **user2** with a subject.

Example

```
EDMMAILQ from=user1@com1.com,to=user2@com2.com,Mesgfile=
c:\report.txt,Subject= "My report"
```

To send mail from **user1@company1.com** to **user2@company2.com** and **user3@company3.com** follow the example below.

Example

```
EDMMAILQ  from=user1@company1.com,to=user2@company2.com;user3@
company3.com,Message=Hello
```

Note: ';' is used as the delimiter between the recipient addresses.

The example below demonstrates how to send a text file, **c:\report.txt**, from **user1** to **user2** with a subject.

Example

```
EDMMAILQ from=user1@com1.com,to=user2@com2.com,Mesgfile=
c:\report.txt,Subject="My report"
```

The example below shows how to send a binary file (**c:\edmwin\ZMASTER.EDM**) and a text file (**c:\report.txt to user2@company2.com**).

Example

```
EDMMAILQ from=user1@company1.com,to=user2@company2.com,Subject=
"Enclosed Files",ATTACH=c:\edmwin\ZMASTER.EDM;c: \report.txt
```

EDMSMTP

EDMSMTP is a TCP/IP Client method that sends e-mail to remote users using SMTP. The method is executed from a command line.

Parameter values can be up to 255 ASCII characters in length, and can contain references to EDM objects and variables. The table below lists the parameters.

Parameter Name	Description
FROM	Required. Specifies the address of the sender (i.e. user1@company1.com).
MESSAGE	Required. Specifies a brief message. The message is limited to 255 characters.
TO	Required. Specifies the recipients of the mail. Multiple users can be listed by using ';' as the delimiter between each recipient (i.e. user1@company1.com;user2@companyb.com).
ATTACH	Optional. Specifies attachment files. Multiple attachments can be listed by using ';' as the delimiter between each attachment (i.e. c:\config.sys;c:\autoexec.bat). Attachments are sent using MIME. Attachments can be binary files.
EDMOBJ	Optional. To specify the EDM object that contains all of the above parameters
MESGFILE	Optional. Specifies the file that contains the message. Is used in place of the MESSAGE parameter if the message is greater than 255 characters.
SUBJECT	Optional. Specifies the subject of the mail.
TIMEOUT	Optional. Specifies the time (seconds) in which a connection must be established with the remote mail server. If the connection cannot be established, an instance of the ZERROR object will be created, and ZBRC will be set to 8 in the ZMASTER object. The default value 60 seconds.

Note: Arguments are expected to be in the **KEYWORD=VALUE** format delimited by commas.

Parameters are used as Keywords and are not case sensitive.

Note: If EDMOBJ is specified, all remaining keywords will be ignored.

Usage

The example below illustrates how to send mail from **user1@company1.com** to **user2@company.com** with a brief message.

Example

```
EDMSMTP from=user1@company.com,to=user2@company2.com,
Message="This is a brief message"
```

Note: The double quote character is needed when a value contains embedded blanks or commas.

The example below illustrates how to send mail from **user1@company1.com** to **user2@company2.com** and **user3@company3.com**.

Example

```
EDMSMTP from=user1@company1.com,to=user2@company2.com;
user3@company3.com,Message=Hello
```

Note: ';' is used as the delimiter between the recipient addresses.

In the example below, a text file is sent from **c:\report.txt** from **user1** to **user2** with a subject.

Example

```
EDMSMTP from=user1@com1.com,to=user2@com2.com,Mesgfile=
c:\report.txt,Subject="My report"
```

The example below shows how to use the timeout parameter. You may want to use this parameter if your are sending e-mail to user on a remote system over a busy network. The timeout specified is 300 seconds.

Example

```
EDMSMTP from=user1@isp.net,to=user2@company.com,timeout=300,
message="Brief Message"
```

The example below shows how to send the following attachments to **user2@company2.com**.

- a binary file: **eg. c:\edmwin\ZMASTER.EDM**,
- a text file, : **C:\report.txt** to

Example

```
EDMSMTP from=user1@company1.com,to=user2@company2.com,Subject=
"Enclosed Files",ATTACH=c:\edmwin\ZMASTER.EDM;c: \report.txt
```

To send mail from a user whose userid is stored in the EDM object ZMASTER in variable ZUSERID, follow the example below.

Example

```
EDMSMTP from=&(ZMASTER.ZUSERID)@company1.com,to=
user2@company2.com,Message=Hello
```

The example below demonstrates how to send mail using an EDM object.

Example

```
EDMSMTP EDMOBJ=TESTOBJ
```

The TESTOBJ should contain the following mandatory variables

- TO
- FROM
- MESSAGE or MESGFILE

For Windows and DOS clients, the Timezone from the ZMASTER object is used.

The variable name for timezone is ZTIMEZON, and it has the following valid values EST, EDT, CST,CDT,MST,MDT,PST,PDT, or the local differential (in hours+minutes) from GMT;, i.e.for example +0500,-0530

If mail cannot be delivered either because the server is busy or the recipient is not a known user on the remote system, an instance is added to the ZERROR object, and the ZBRC variable is set to 8 in the ZMASTER object. This variable indicates that the method failed.

ZERRORM

ZERRORM is a debugging feature that tracks and provides meaningful messages for events that occur during an EDM Client Connect session.

When this feature is used, an in-storage object called ZERRORM is created for each task. After an EDM Client Connect session, clients can download the object so they can analyze the messages contained with in it, and correct problems that occurred.

The events tracked by ZERRORM are documented in the Manager Log. The Manager Log contains thousands of messages that track all of the users connected to the EDM Manager. Through ZERRORM, EDM Clients receive messages that are directly related to them.

Currently, there are 14 ZERRORM events.

A The EDM Manager Method Reference

This appendix is a reference for EDM Manager methods.

Version 2.X Method Names

Version 2.X Name	Version 3.X Name
NEW	EDMMAILQ
NEW	EDMMPCHK
NEW	EDMMPUSH
EDMSIGN	EDMMSIGN
EDMSIGNR	EDMMSINR
EDMMULOG	EDMMULOG
ZDCLASS	EDMMDCLA
ZDELINS	EDMMDINS
ZDELOBJS	EDMMDOBJ
ZDELPROF	EDMMDPRO
ZEXIST	EDMMEXIS
ZGETPROF	EDMMGPRO
ZIRXALOC	EDMMALLO
ZNOTIFY	EDMMNFYT
ZOBJCMPR	EDMMCMPR
ZOBJCOPY	EDMMCOPY
ZOBJDELI	EDMMDELI
ZOBJDELV	EDMMDELV
ZOBJSORT	EDMMSORT
ZPROMANY	EDMMPROM
ZPUTHIST	EDMMPHIS
ZPUTPROF	EDMMPPRO
ZSIMRESO	EDMMRESO
ZTOUCH	EDMMTUCH
ZUPDSZ	EDMMUPSZ
ZVARDEL	EDMMVDEL
ZVARGBL	EDMMVGBL
ZVARLOG	EDMMOLOG
ZXREF	EDMMXREF

Method Naming Standards

The names of the EDM Manager methods provided by EDM have been changed as of the release of EDM Version 3.x. These method names were standardized to make it easier for you to manage methods at your site.

For example, the ZDELOBJS method is now named EDMMDOBJ. Note that all the methods named by using the method naming standards enforced as of Version 3 are structured similarly to EDMMDOBJ, as follows:

EDMMDOBJ

EDM identifies the method as an EDM method

M identifies the method as an EDM Manager method

The last four characters denote the function for which the method can be employed.

The following table provides you with a list of methods, including both the Version 2 and Version 3 method names, as well as a brief description.

> **Note**: EDM continues to support the EDM Manager methods for both the Version 2 and the Version 3 method names. That is, both the ZDELOBJS and EDMMDOBJ continue to be supported.
>
> You should be aware that the method names have been changed. And, as you begin to work within the Version 3.x environment, we suggest that you employ the newly named methods.

EDM Manager Methods

Version 3.X Name	Version 2.X Name	Description
EDMMAILQ	NEW	Deposits e-mail in the Mail Queue so it can be sent to a remotes system user
EDMMALLO	ZIRXALOC	Allocates an external data set (MVS only)
EDMMCMPR	ZOBJCMPR	Compresses an in-storage object
EDMMCOPY	ZOBJCOPY	Copies an in-storage object
EDMMDCLA	ZDCLASS	Deletes a class from the EDM database
EDMMDELI	ZOBJDELI	Deletes and instance from an in-storage object
EDMMDELV	ZOBJDELV	Deletes a variable from all instances of an in-storage object
EDMMDINS	ZDELINS	Deletes an instance or instances from within a class in the EDM database
EDMMDOBJ	ZDELOBJS	Deletes an in-storage object
EDMMDPRO	ZDELPROF	Deletes an object in the PROFILE database
EDMMEXIS	ZEXIST	Verifies the existence of a given class or instance in the EDM database
EDMMGPRO	ZGETPROF	Creates a global object from the PROFILE database.
EDMMNFYT	ZNOTIFY	Executes a PUSH on an EDM Client.
EDMMOLOG	ZVARLOG	Displays the contents of an in-storage object.
EDMMPCHK	NEW	Check an object for duplicate resources before it is promoted.
EDMMPHIS	ZPUTHIST	Puts an in-storage object into the HISTORY database.
EDMMPPRO	ZPUTPROF	Puts an in-storage object into the PROFILE database.
EDMMPROM	ZPROMANY	Adds an instance to the EDM database.

Version 3.X Name	Version 2.X Name	Description
EDMMPUSH		
EDMMRESO	ZSIMRESO	Resolves specified objects.
EDMMSIGN	EDMSIGN	Authenticates users against the EDM database
EDMMSINR	EDMSIGNR	Authenticates users against external security systems
EDMMSORT	ZOBJSORT	Sorts instances, by stems, of in-storage objects
EDMMTUCH	ZTOUCH	Updates the date/time stamp of an instance
EDMMULOG	EDMMULOG	Used to write a new log file.
EDMMUPSZ	ZUPDSZ	Compares the size of a ZRSOURCE instance to the size of equivalent instances in the RESOURCE database
EDMMVDEL	ZVARDEL	Deletes all in-storage objects
EDMMVGBL	ZVARGBL	Migrates values from one in-storage object to another and deletes the source object.
EDMMXREF	ZXREF	Cross references class and instance usage during the object resolution process

Using EDM Manager Methods

In EDM, a method is a program or procedure that can be packaged and exchanged as an object, specifically as an instance of the ZMETHOD class. By connecting a ZMETHOD class instance to another class instance, you can specify where and when that procedure will run. Use the instance screen associated with the method to connect and initiate the method

You can also run an ASM method from a REXX script, enabling you to execute methods outside of the object resolution process. The following is an example of the format used to execute a method in this way.

```
ADDRESS EDMLINK EDMMCMPR 'ZTEST    '
```

EDM Manager methods allow you to manipulate EDM objects and database components at the system (Manager) level as opposed to the client or individual workstation. EDM objects, referred to as in-storage objects, are those used or created during the object resolution process. Database components are those entities (files, domains, classes, instances, and variables) that reside in the EDM database.

The EDM Manager methods that affect EDM objects and database entities are as follows:

Method Name	EDM Object	Database Entity
EDMMAILQ	✓	N/A
EDMMALLO	N/A	N/A
EDMMCHK	N/A	N/A
EDMMCMPR	✓	
EDMMCOPY	✓	
EDMMDCLA		✓
EDMMDELI	✓	
EDMMDELV	✓	
EDMMDINS		✓
EDMMDOBJ	✓	
EDMMDPRO		✓
EDMMEXIS		✓
EDMMGPRO		✓
EDMMNTFY	N/A	N/A
EDMMOLOG	✓	
EDMMPCHK	N/A	N/A
EDMMPHIS	✓	
EDMMPRO	✓	
EDMMPROM		✓
EDMMPUSH	N/A	N/A
EDMMRESO	✓	
EDMMSIGN	N/A	N/A
EDMMSINR	N/A	N/A
EDMMSORT	✓	
EDMMTUCH		✓
EDMMULOG	N/A	N/A
EDMMUPSZ		✓
EDMMVDEL		✓

Method Name	EDM Object	Database Entity
EDMMVGBL		✓
EDMMXREF	✓	

Methods are often used in conjunction with other methods to achieve a specific purpose. For example, you can use the EDMMDOBJ method to delete an in-storage object, then execute EDMMCOPY to copy an object, giving it the original object name.

Methods must be connected to other class instances at an appropriate point to achieve the desired result. For example, you do not want to delete an instance before it is used in object resolution, or create an instance when it will be overwritten.

The default file and domain used by some methods can be specified in the DBASE and DOMAIN values of the MGR_STARTUP member/setting of the PARMLIB, .edmprof , or EMDPROF.DAT files.

EDMMAILQ

Method Name EDMMAILQ

Description Deposits e-mail in the Mail Queue (OutBox). The [MGR_SMTP_MAIL] section must be added to the EDMPROF.DAT file for this method to execute correctly.

Usage This method can be used to place e-mail in the Mail Queue so the SMTP Send Manager (zsmtsmgr) can send it to a user on a remote system. If mail is to be delivered to a remote user, the zsmtsmgr task must be included in the [MGR_ATTACH_LIST] section of the EDMPROF.DAT file. See Chapter 2, "*Tuning the EDM Manager for Windows NT*".

Type EDM Manager

Old Name New

Platform MVS, Solaris, HP-UX, AIX, OS/2, Windows NT.

Parameters `FROM, MESSAGE, TO, ATTACH, MESGFILE, SUBJECT.`

Parameter Name	Description
FROM	Required. The sender's address (user1@company1.com).
MESSAGE	Required. Specifies a brief message (limited to 255 characters).
TO	Required. Specifies the E-mail recipients. Multiple users can be listed by using ';' as the delimiter between each recipient. (user1@company1.com;user2@companyb.com).
ATTACH	Optional. Specifies attachment files. Multiple attachments can be listed by using ';' as the delimiter between each attachment (c:\config.sys;c:\autoexec.bat). Attachments are sent using MIME. Attachments can be in binary format.
MESGFILE	Optional. Specifies the file that contains the message. Is used in place of the MESSAGE parameter if the message is greater than 255 characters.
SUBJECT	Optional. Specifies the subject of the e-mail.

Note: Arguments are expected to be in the **KEYWORD=VALUE** format delimited by commas.

Parameters are used as Keywords and are not case sensitive.

In the example below mail is sent from **user1@company1**.com to **user2@company.com** with a brief message.

Example

```
EDMMAILQ from=user1@company.com,to=user2@company2.com,Message=
"This is a brief message"
```

Note: The double quote character is needed when a value contains embedded blanks or commas.

In the example below, a text file (**c:\report.txt**) is sent from **user1** to **user2** with a subject.

Example

```
EDMMAILQ from=user1@com1.com,to=user2@com2.com,Mesgfile=
c:\report.txt,Subject= "My report"
```

Possible MVS Return Codes

Return Code	Description
0	Method was successful.
8	Error, no input parameters.
8	An error was detected, ZVCB is unavailable.
8	Object database access error.

Possible Non-MVS Return Codes

Return Code	Description
0	The method was successful.
8	An error was detected, the method failed.

EDMMALLO

Method Name EDMMALLO

Description This method enables you to allocate a file dynamically for use in a REXX exec. The file will be freed automatically after the EXECIO command is issued with the "FINIS" option, or when the EDMMALLO method is issued. The maximum number of files a single REXX exec can allocate is 50.

Usage This method can be used to set aside storage for the resulting output of a REXX method or program.

Type EDM Manager

Old Name ZIRXALOC

Platform MVS

Parameters DSN, MEM, DISP

Parameter Name	Data Set Name to be Allocated
DSN	Concatenated to the fully qualified data set name of file to be allocated.
DISP	Concatenated to the file disposition. "DISP=SHR" is recommended.
MEM	Concatenated to the member name. This should only be specified if the file is a Partitioned Data Set (PDS) or PDS/E.

Example

```
/************************* REXX ********************/

SAY 'SAMPLE FILE ALLOCATION'

/************ ALLOCATE THE DATA SET ******************/

DSN = 'DSN=USER1.EDM.TESTLIB';

DISP= 'DISP=SHR';

MEM = 'MEM=TESTMEM';

/*****************************************************/

/* ALLOCATE THE FILE                                 */

/*****************************************************/

DDN = EDMMALLO(DSN,MEM,DISP)

/*********************************************************/

/* READ ALL THE RECORDS FROM THE FILE INTO A STRUCTURE     */

/* WITH RECS. AS THE STEM VARIABLE FOR EACH RECORD.        */
```

```
/* THE FINIS OPTION WILL CAUSE THE DATA SET TO BE CLOSED   */
/* AND DEALLOCATED. IF THE FINIS OPTION IS NOT SPECIFIED   */
/* REXX KEEPS THE DATA SET OPEN AND THE "EXECIO 0" COMMAND */
/* MUST BE USED TO CLOSE THE FILE.                         */
/*********************************************************/
"EXECIO * DISKR" DDN "(STEM RECS. FINIS"
SAY 'ALLOCATED   DDN = ' DDN;
/*"EXECIO * DISKR" DDN "(STEM RECS. " <---THIS WOULD NOT CLOSE/FREE */
/*********************************************************/
/* DE-ALLOCATE THE FILE. THIS IS NOT NECESSARY AND WILL   */
/* FAIL IF THE DATA SET ALREADY FREED. IT SHOULD BE USED  */
/* IN CASE AN ERROR IN THE REXX EXEC DOES NOT EXECUTE     */
/* THE EXECIO COMMAND AT ALL, NEVER OPENING/CLOSING AND   */
/* FREEING THE FILE.IF THE DATA SET IS NOT FREED OTHER    */
/* USERS MAY NOT BE ABLE TO ACCESS THE DATA SET UNTIL THE */
/* EDM MANAGER IS TERMINATED.                             */
/*********************************************************/
DDNF = EDMMDALO(DDN);
/*********************************************************/
/* ALLOCATE THE FILE                                      */
/*********************************************************/
  SAY 'DEALLOCATED DDN = ' DDF;
END
SAY '.> NFYT ends ==================>';
```

Possible MVS Return Codes

Return Code	Description
0	The method was successful.

Possible Non-MVS Return Codes

Return Code	Description
0	The method was successful.
8	An error was detected, the method failed.

EDMMCMPR

Method Name EDMMCMPR

Description Compresses an in-storage object.

Usage This method can be used to remove gaps in the database storage after variables have been deleted with EDMMDELV, or other database manipulation.

Type EDM Manager

Old Name ZOBJCMPR

Platform MVS, Solaris, HP-UX, AIX, OS/2, Windows NT.

Parameters `object`

Parameter Name	Description
object	The name of the in-storage object to be compressed.

Example

```
ADDRESS EDMLINK EDMMCMPR 'ZTEST   ' ;
```

Possible MVS Return Codes

Return Code	Description
0	The method was successful.
16	The compressed object was not found.
16	An error was detected, storage is unavailable.

Possible Non-MVS Return Codes

Return Code	Description
0	The method was successful.
8	An error was detected, the method failed.

EDMMCOPY

Method Name EDMMCOPY

Description Copies an in-storage object. The resulting object has the same variables and number of heaps as the original object.

Usage This method can be used to create a backup of an object.

Type EDM Manager

Old Name ZOBJCOPY

Platform MVS, Solaris, HP-UX, AIX, OS/2, Windows NT.

Parameters `fromobject, toobject`

Parameter Name	Description
fromobject	The name of the existing in-storage object to be copied.
toobject	The name of the new in-storage object to be created.

Example

```
ADDRESS EDMLINK EDMMCOPY 'OBJECT1,OBJECT2' ;
```

Possible MVS Return Codes

Return Code	Description
0	The method was successful, or nothing happened.
16	An error was detected, the method failed. The old object could not be found
16	An error was detected, a new name exists.
16	An error was detected, storage is unavailable.
16	An error was detected, $PVTQ failed.
16	An error was detected, invalid name.
16	An error was detected, invalid parameters

Possible Non-MVS Return Codes

Return Code	Description
0	The method was successful.
8	An error was detected, the method failed.

EDMMDCLA

Method Name EDMMDCLA

Description Deletes a class and all associated instances from the EDM database. Note that EDMMDCLA will not delete ZRSOURCE instances.

Usage This method can be used to customize your database or to migrate a database from a test to a production environment.

Type EDM Manager

Old Name ZDCLASS

Platform MVS, Solaris, HP-UX, AIX, OS/2, Windows NT.

Parameters **Non-MVS:** `domain,class`

MVS: `domain--class---`

Parameter Name	Description
domain	The name of the domain that contains the class to be deleted. A maximum of 8 characters.
class	The name of the class to be deleted. A maximum of 8 characters.

Example

```
/****************************** REXX ***************/

';

 DOMAIN = 'SYSTEMX ';

 CLASS  = 'TESTCLAS';

PARM = SUBSTR(DOMAIN,1,8)  ||  SUBSTR(CLASS,1,8);

 SAY 'PARM STRING IS ' PARM;

 ADDRESS EDMLINK EDMMDCLA PARM;
```

Possible MVS Return Codes

Return Code	Description
0	The method was successful.
8	An error was detected, there are no input parameters.
8	An error was detected, ZVCB is unavailable.
8	An error was detected, object database access error.

Possible Non-MVS Return Codes

Return Code	Description
0	The method was successful.
8	An error was detected, the method failed.

EDMMDELI

Method Name EDMMDELI

Description Deletes an instance of an in-storage object.

Usage This method can be used, during testing, to delete redundant or unnecessary instances.

Type EDM Manager

Old Name ZOBJDELI

Platform MVS, Solaris, HP-UX, AIX, OS/2, Windows NT.

Parameters `object, instance#`

Parameter Name	Description
object	The name of the in-storage object to delete an instance from.
instance#	The instance number to delete.

Example

```
/************************* REXX ********************/

      DPARM = 'TESTOBJ,'||1|| ' ';

      ADDRESS EDMLINK EDMMDELI DPARM;
```

Possible MVS Return Codes

Return Code	Description
0	The method was successful.
8	An error was detected, instance not found.
12	An error was detected, object not found.
16	An error was detected, invalid instance.
28	An error was detected, invalid parameters.

Possible Non-MVS Return Codes

Return Code	Description
0	The method was successful.
8	An error was detected, the method failed.

EDMMDELV

Method Name EDMMDELV

Description Deletes a variable from all instances of an in-storage object. The EDMMDELV method verifies the existence of the input object name and finds the specified variable in that in-storage object. The variable value is then removed from each instance in the in-storage object.

Usage This method can be used for testing or problem resolution.

Type EDM Manager

Old Name ZOBJDELV

Platform MVS, Solaris, HP-UX, AIX, OS/2, Windows NT.

Parameters `object, variable`

Parameter Name	Description
object	The object that contains the variable to be deleted.
variable	The name of the variable to be deleted.

Example

```
/************************* REXX ********************/


ADDRESS EDMLINK EDMMDELV 'TESTOBJ,VAR00001' ;

SAY 'QAREXX ****** VAR00001 DELETED FROM OBJECT TESTOBJ' ;
```

Possible MVS Return Codes

Return Code	Description
0	The specified variable was deleted.
16	An error was detected, the object could not be found.
16	An error was detected, the variable could not be found.
16	An error was detected, storage is unavailable.
28	An error was detected, invalid parameter.

Possible Non-MVS Return Codes

Return Code	Description
0	The method was successful.
8	An error was detected, the method failed.

EDMMDINS

Method Name EDMMDINS

Description Deletes or displays an instance, or range of instances, within a class from the database. This method permits the use of wildcards (*). Note that the displayed instances will be written to the EDM Manager Log even if all other TRACE settings are off.

Usage EDMMDINS can be used for database maintenance or to collect information associated with testing or problem resolution.

Type EDM Manager

Old Name ZDELINS

Platform MVS, Solaris, HP-UX, AIX, OS/2, Windows NT.

Parameters Non-MVS: `file,domain,class,option,frominst, toinst`

MVS: `file----domain—class---option-frominst---(32)toinst---(32)`

Parameter Name	Description
file	The file of the instance(s) to be displayed/deleted.
domain	The domain of the instance(s) to be displayed/deleted.
class	The class of the instance(s) to be displayed/deleted.
option	The word “DISPLAY” if instance(s) is to be displayed, “DELETE” if instance(s) is to be deleted.
frominst	The instance name or starting name to be deleted or displayed.
toinst	The instance name to be deleted or displayed. Blanks in this field would indicate that it is a single instance to display or delete and not a range.

Example

```
/************************* REXX ********************/

FILE = 'PRIMARY'

 DOMAIN = 'SYSTEMX'

 CLASS = 'ZRSOURCE';

 FROMIN = 'TSO_                ;

 TOINS = '                ;

 OPTION = 'DISPLAY';

 PARM = FILE||DOMAIN||CLASS||OPTION||FROMIN||TOINS;

 SAY 'PARM STRING IS 'PARM;

 ADDRESS EDMLINK EDMMDELI PARM;
```

Possible MVS Return Codes

Return Code	Description
0	No errors were detected, the method was successful.
8	An error was detected, no input parameter.
8	An error was detected, ZVCB is unavailable.
8	An error was detected, object database access error.

Possible Non-MVS Return Codes

Return Code	Description
0	The method was successful.
8	An error was detected, the method failed.

EDMMDOBJ

Method Name EDMMDOBJ

Description Deletes an in-storage object.

Usage This method can be used to for testing and problem resolution.

Type EDM Manager

Old Name ZDELOBJS

Platform MVS, Solaris, HP-UX, AIX, OS/2, Windows NT.

Parameters `object`

Parameter Name	Description
object	The name of the in-storage object to be deleted.

Example

```
ADDRESS EDMLINK EDMMDOBJ 'ZTEST';
```

Possible MVS Return Codes

Return Code	Description
0	The specified object was deleted.

Possible Non-MVS Return Codes

Return Code	Description
0	The method was successful.
8	An error was detected, the method failed.

EDMMDPRO

Method Name EDMMDPRO

Description Deletes an object in the PROFILE database.

Usage This method can be used for database maintenance or after simulation.

Type EDM Manager

Old Name ZDELPROF

Platform MVS, Solaris, HP-UX, AIX, OS/2, Windows NT.

Parameters Non-MVS: `domain,class`

MVS: `domain--class---`

Parameter Name	Description
domain	The domain the object to be deleted is located in.
class	The class the object to be deleted is located in.

Possible MVS Return Codes

Return Code	Description
0	The specified object was deleted.
8	An error was detected, ZVCB is unavailable.
16	An error was detected, profile object oriented database is unavailable.
28	An error was detected, invalid parameters.

Possible Non-MVS Return Codes

Return Code	Description
0	The method was successful.
8	An error was detected, the method failed.

EDMMEXIS

Method Name EDMMEXIS

Description Verifies the existence of a given class or instance in the EDM object database.

Usage This method can be used to verify that a new user (class or instance) has successfully been created, and is present in the EDM database, for testing or reporting purposes.

Type EDM Manager

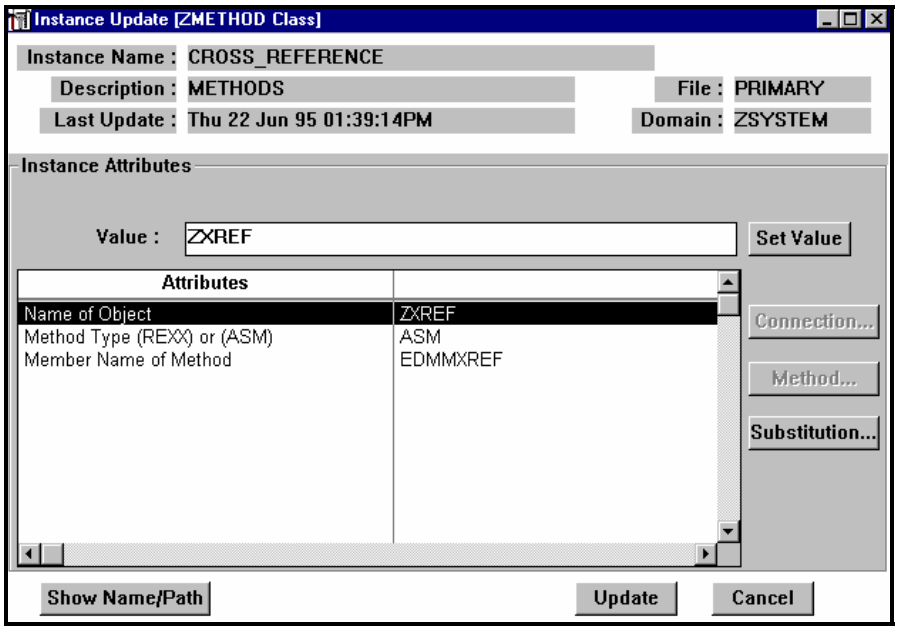
Old Name ZEXIST

Platform MVS, Solaris, HP-UX, AIX, OS/2, Windows NT.

Parameters Non-MVS: `file,domain,type,class,instance,`

MVS: `file(4)domain(8)type(1)class(8) instance(32)`

Parameter Name	Description
file	The file that contains the class or instance to be verified.
domain	The domain that contains the class or instance to be verified.
type	The type of file.
class	The class that contains the instance or class record to be verified.
instance	The instance to be verified.



UNIX Parameters

Parameter Name	Description
file (optional)	The file that contains the class or instance to be verified. The default is the current database file.
domain (optional)	The domain that contains the class or instance to be verified. The default is the current database domain.
class	The class that contains the instance or class record to be verified.
instance	The instance to be verified.

MVS Example

```
/*********************** EDMMEXIS ******************/

/* MVS Example  */

CORR = 'PRIM' ;

DOMAIN = 'SYSTEMX ' ;

TYPE = 'I' ;

CLASS = 'USER    ' ;

INST = 'USER1   ' ;

PARM = SUBSTR(CORR,1,5) || SUBSTR(DOMAIN,1,8) || SUBSTR(TYPE,1,1) ,

       SUBSTR(CLASS,1,8) || SUBSTR(INST,1,32) ;

ADDRESS EDMLINK EDMMEXIS PARM ;

IF RC = 0 THEN

    SAY 'QAREXX ****** OBJECT ' INST ' EXISTS;
```

UNIX Example

```
/****************************** EDMMEXIS  ************************/

/* UNIX Example  */

FILE = 'PRIMARY'

DOMAIN = 'SYSTEMX ' ;

CLASS = 'USER' ;

INST = 'USER1' ;

PARM = FILE  || '.' || DOMAIN || '.' || CLASS || '.' || INST ;

ADDRESS EDMLINK EDMMEXIS PARM ;

IF RC = 0 THEN
```

```
SAY 'QAREXX ****** OBJECT ' INST ' EXISTS;
```

Possible MVS Return Codes

Return Code	Description
0	The specified class/instance exists.
8	An error was detected, connection name format error.
8	An error was detected, the instance name is missing or invalid.
8	An error was detected, instance name length error.
8	An error was detected, ZVCB is unavailable.
8	An error was detected, object database access error.

Possible Non-MVS Return Codes

Return Code	Description
0	The method was successful.
8	An error was detected, the method failed.

EDMMGPRO

Method Name EDMMGPRO

Description This method creates an in-storage object from a Profile database object. The EDMMGPRO method gets the dbobject object from the Profile database and puts it in storage as inobject. The domain in the Profile database is the userid "ZUSERID", which is found in the current object or in the ZMASTER object. If ZUSERID is not found, the object is put in the __UNKNOWN domain.

Usage This method could be used to retrieve an object, i.e. ZMASTER, create a duplicate of the object, i.e. called ZTEST, and to save the duplicate in the EDM database

Type EDM Manager

Old Name ZGETPROF

Platform MVS, Solaris, HP-UX, AIX, OS/2, Windows NT.

Parameters `dbobject, inobject, ZUSER, instance`

Parameter Name	Description
dbobject	The Profile database object name.
inobject	The name of the in-storage object to be created.
ZUSER (optional)	The name of the ZUSER Domain
instance (optional)	The name of the instance.

Example

```
/******************** REXX ********************************/

    PARM='ZSTATUS,ZSTATUS,'ZMASTER.ZUSERID;

    ADDRESS EDMLINK ZGETPROF PARM; /* GET OLD PROFILE.?.ZSTATUS*/
```

Possible MVS Return Codes

Return Code	Description
0	The method was successful.
8	An error was detected, profile not found.
8	An error was detected, ZUSERID not found.
8	An error was detected, domian not found.
16	An error was detected, RODD error.
16	An error was detected, instance missing.
20	An error was detected, profile not open for user.
24	An error was detected, ZVCB unavailable.
28	An error was detected, parm error.

Possible Non-MVS Return Codes

Return Code	Description
0	The method was successful.
8	An error was detected, the method failed.

EDMMNFYT

Method Name EDMMNFYT

Description EDM Notify enables you to initiate a PUSH (the execution of a program or programs on an EDM Client desktop from another location). To execute EDM Notify successfully, EDMEXECD must be running on the EDM Clients you are executing a PUSH on.

Usage An EDM Manager can use this method to notify EDM Clients to initiate the EDM Client Connect Process.

Type EDM Manager

Old Name ZNOTIFY

Platform MVS, Solaris, HP-UX, AIX, OS/2, Windows NT, Windows 3.1, Windows 95.

Parameters `target IP address, port, UserID, password, "process to run"`

Parameter Name	Description
target IP address	The IP address of the EDM Client desktop you are executing a PUSH on.
port	For MVS only. The port number. Should have the same value as the ZMASTER port number.
userID	The EDM Client user ID.
password	The ZPWD for the target terminal's ZMASTER object.
"process to run"	The application you are forcing the EDM Client desktop to execute.

MVS Example

```
ADDRESS EDMLINK EDMMNFYT ';
```

In MVS, you will configure NFYTTST in EDM Console.

When sending from UNIX to:

Platform	Example
UNIX	`EDMMNFYT 204.7.83.58 USERID ZPWD "c:\ msoffice\msword.exe"`
OS/2	`EDMMNFYT 204.83.58 "c:\ msoffice\msword.exe"`
MAC	`EDMMNFYT 204.7.83.58 USERID "MACDRIVE:Microsoft:Word"`
Windows, WinNT, Win95	`EDMMNFYT 204.83.58 "c:\ msoffice\msword.exe"`

When sending from OS/2 to:

Platform	Example
UNIX	`EDMMNFYT 204.7.83.58 -1 USERID -p ZPWD "c:\ msoffice\msword.exe"`
OS/2, MAC, Windows, WinNT, Win95	`EDMMNFYT 204.83.58 "c:\ msoffice\msword.exe"`

When sending from Windows NT to:

Platform	Example
UNIX	`EDMMNFYT 204.7.83.58 "c:\ msoffice\msword.exe"`
OS/2, MAC, Windows, WinNT, Win95	`EDMMNFYT 204.83.58 "c:\ msoffice\msword.exe"`

On the UNIX server, to ensure that the UNIX process was started, type the following command at the UNIX prompt:

```
ps -u [username]
```

Possible MVS Return Codes

Return Code	Description
0	The method was successful.
16	An error was detected, object not found.

Possible Non-MVS Return Codes

Return Code	Description
0	The method was successful.
8	An error was detected, the method failed.

EDMMOLOG

Method Name EDMMOLOG

Description Writes the contents of an in-storage object in the EDM Manager log. Note that EDMMOLOG will write to EDM Manager Log even if all other TRACE settings are off.

Usage This method can be used for reporting, to verify that an expected process has actually taken.

Type EDM Manager

Old Name ZVARLOG

Platform MVS, Solaris, HP-UX, AIX, OS/2, Windows NT.

Parameters `object`

Parameter Name	Description
object	The name of the in-storage object to be displayed.

Example

```
ADDRESS EDMLINK EDMMOLOG 'ZMASTER';
```

Possible MVS Return Codes

Return Code	Description
0	The method was successful.
16	An error was detected, object not found.

Possible Non-MVS Return Codes

Return Code	Description
0	The method was successful.
8	An error was detected, the method failed.

EDMMPCHK

Method Name EDMMPCHK

Description Performs a pre-promote check on an object for duplicate resources. If duplicate resources are found, selected resource versions, or all resource versions can be promoted.

Usage This method can be used to scan an object for duplicate instances before it's promoted.

Type EDM Manager

Old Name

Platform MVS, Solaris, HP-UX, AIX, OS/2, Windows NT.

Parameters

Parameter Name	Description

Possible MVS Return Codes

Return Code	Description
0	The method was successful.
8	An error was detected, object not found.
8	An error was detected, object database access error.
8	An error was detected, database error.
8	An error was detected, ZVCB is unavailable.
8	An error was detected, object not found.

Possible Non-MVS Return Codes

Return Code	Description
0	The method was successful.
8	An error was detected, the method failed.

EDMMPHIS

Method Name EDMMPHIS

Description Puts an in-storage object into the HISTORY file database. The EDMMPHIS method takes the “inobject” in storage, and puts it in the HISTORY file as dbobject. The domain used in the HISTORY file is the DATE/TIME stamp found in “current” object, or in ZMASTER object. If userid is not found, the object is put in the “_UNKNOWN” domain.

Usage This method can be used for problem resolution, fine-tuning EDM Manager operations, or to document a specific condition or event.

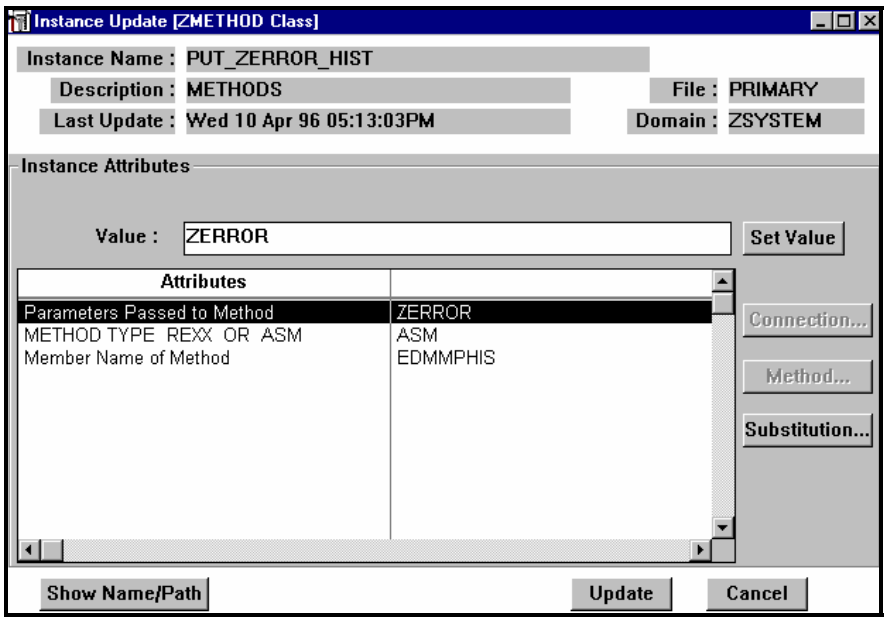
Type EDM Manager

Old Name ZPUTHIST

Platform MVS, Solaris, HP-UX, AIX, OS/2, Windows NT.

Parameters `inobject, dbobject`

Parameter Name	Description
inobject	The object name of the in-storage object.
dbobject (optional)	The object name that will be put in the History file.



Example

```
ADDRESS EDMLINK EDMMPHIS 'ZCOMPARE,ZCOMPARE';

ADDRESS EDMLINK EDMMPHIS 'ZSTATUS';
```

Possible MVS Return Codes

Return Code	Description
0	The method was successful.
8	An error was detected, object not found.
8	An error was detected, object database access error.
8	An error was detected, database error.
8	An error was detected, ZVCB is unavailable.
8	An error was detected, object not found.

Possible Non-MVS Return Codes

Return Code	Description
0	The method was successful.
8	An error was detected, the method failed.

EDMMPPRO

Method Name EDMMPPRO

Description Puts an in-storage object into the Profile database. The EDMMPPRO method takes the in storage "inobject" and puts it in the Profile file as "dbobject". The domain in the Profile file is "ZUSERID", found in the "inobject" or in the ZMASTER object. If ZUSERID is not found, the "dobject" is put in the "_UNKNOWN" domain.

Usage This method can be used to collect information on relevant events for testing or reporting purposes.

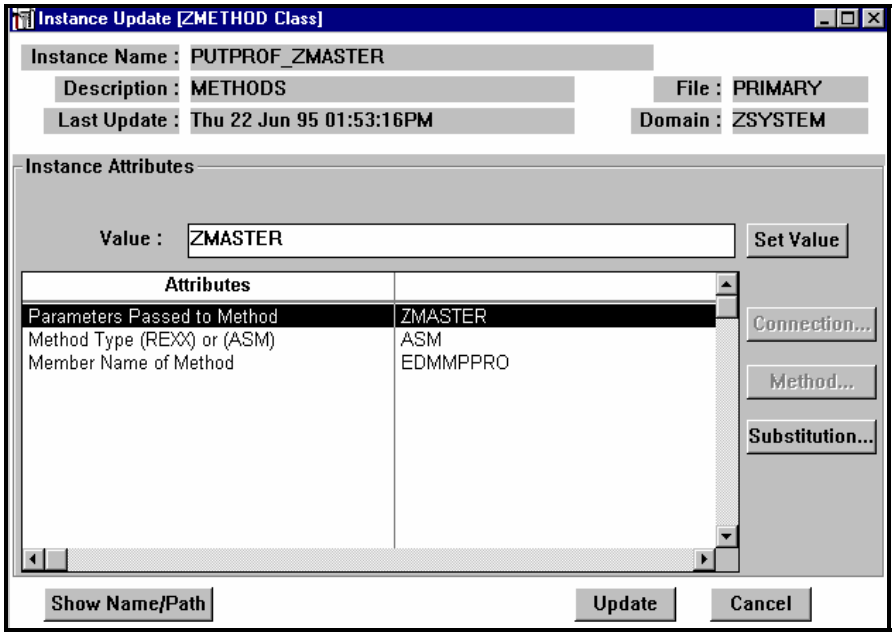
Type EDM Manager

Old Name ZPUTPROF

Platform MVS, Solaris, HP-UX, AIX, OS/2, Windows NT.

Parameters `inobject, dbobject`

Parameter Name	Description
inobject	The name of the in-storage object.
dbobject (optional)	The object name that will be put in Profile database. If dbobject is not specified, the default is the object name.



Example

```
/******************** REXX ********************************/
```

```
ADDRESS EDMLINK ZPUTPROF 'ZCOMPARE,ZCOMPARE ';
```

```
ADDRESS EDMLINK ZPUTPROF 'ZSTATUS ';
```

Possible MVS Return Codes

Return Code	Description
0	The method was successful.
8	An error was detected, the template object could not found.
8	An error was detected, ZUSERID not found.
8	An error was detected, ZVCB is unavailable.
12	An error was detected, the object could not be found.
16	An error was detected, ENQ/DEQ error.
16	An error was detected, profile database error.
20	An error was detected, object database access error.
28	An error was detected, “FROM-TO” not valid.

Possible Non-MVS Return Codes

Return Code	Description
0	The method was successful.
8	An error was detected, the method failed.

EDMMPROM

Method Name EDMMPROM

Description The EDMMPROM method allows you to add instances to and delete instances from the EDM Manager.

This method allows you to add and update instances in the Primary and Secondary files based on the object you specify as a parameter. The object specifies which file, domain, class, and instance will be added/updated by specifying four variables: ZADMFILE,"ZADMDOMN",ZADMCLAS, and ZADMINST. Each instance in "object" can specify an instance to be added/updated. The variables in each instance contain the values to add/update in the database. Any variable found in the target instance will be set. Any variable not found in the target instance will be ignored. Correct length adjustment is performed, including blank padding and truncation.

Usage You can use this method to automatically define class instances in any existing EDM class. In addition, the following conditions apply to the EDMMPROM method:

You cannot use the EDMMPROM method to dynamically add a file, domain, or class; these entities must exist. Also, any fields being processed must already be defined in the target class, EDMMPROM will not modify classes.

- Any field type can be processed, including variables, classes, and methods.
- Connect and method fields that may have duplicate names can be manipulated using commas.

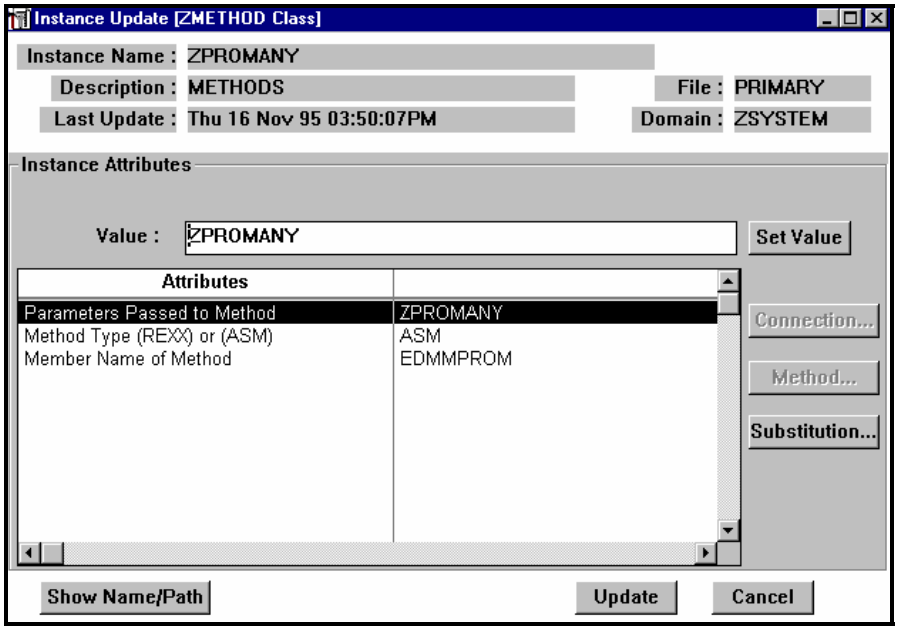
Type EDM Manager

Old Name ZPROMANY

Platform MVS, Solaris, HP-UX, AIX, OS/2, Windows NT.

Parameters `object`

Parameter Name	Description
object	The name of the in-storage object.



Possible MVS Return Codes

Return Code	Description
0	All return codes for this method are zero.

Possible Non-MVS Return Codes

Return Code	Description
0	The method was successful.
8	An error was detected, the method failed.

EDMMPUSH

Method Name EDMMPUSH

Description Receives input requests, and inserts them into queues processed by the notify manager.

Usage This method can be used to enhance EDMNOTIFY by receiving requests from objects that are sent to the EDM Manager, and queuing them.

You can automate the usage of EDMMPUSH by creating an instance for it in PRIMARY.ZSYSTEM.ZMETHOD, and connecting it to either PRIMARY.ZSYTEM.ZPROCESS.ZMASTER or PRIMARY.ZSYSTEM.ZPROCESS.ZADMIN.

Type EDM Manager

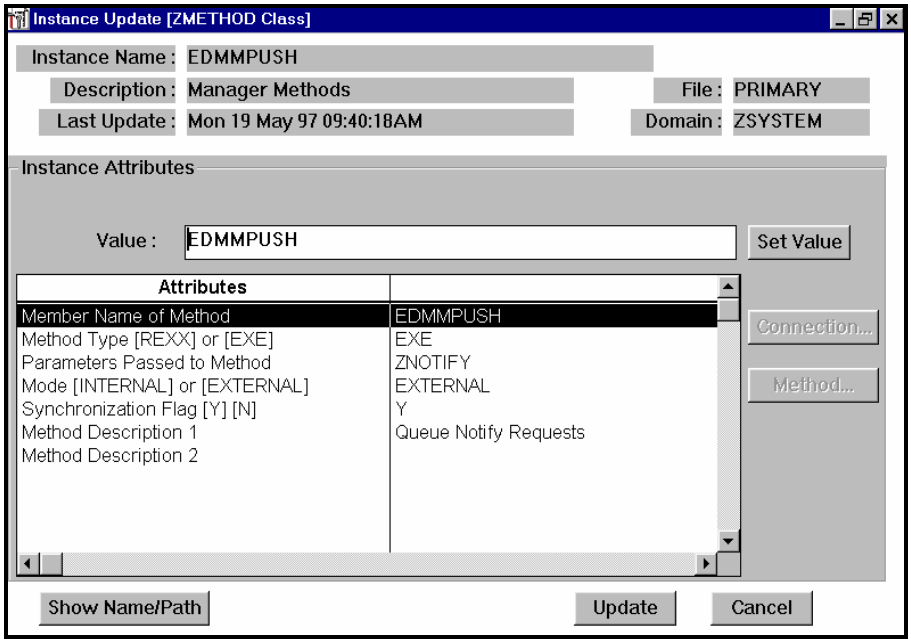
Old Name New

Platform AIX, OS/2, Windows NT.

Parameters `object name, default object = ZNOTIFY`

Parameter Name	Description
object name	Specifies the name of the in-bound object to retrieve information from.
NFTYPE	Specifies the communication protocol. Only TCP/IP is supported at this time.
NFYPROC	Controls the processing of a current heap request. Yes or No. Default is Yes.

Example



Possible Non-MVS Return Codes

Return Code	Description
0	The method was successful.
8	An error was detected, the method failed.

EDMMRESO

Method Name EDMMRESO

Description Resolves the object specified by DomainClassInst(Msg), and the resulting objects are left in storage. Any prerequisite objects needed by that resolution must already have been built in storage. For example, for USER.&ZUSERID resolution, a ZMASTER object might have to be constructed containing ZUSERID, ZOS variables, otherwise, the resolution may not be entirely successful.

Usage You can use this method to automatically define class instances in You can use the EDMMRESO method to simulate the EDM Client connect Process, EDM Reporting or to verify new EDM Client schema verification. All objects are left in-storage so, successive iterations could accumulate potentially large objects. This can be used for testing large resolutions. In addition, EDMMRESO can be used to initiate secondary resolutions based on certain conditions.

Type EDM Manager

Old Name ZSIMRESO

Platform MVS, Solaris, HP-UX, AIX, OS/2, Windows NT.

Parameters MVS: `domain(8)class(8)instance(32) message(8)defaultobject(8)'`

Unix: 'file domain class instance message'

Parameter Name	Description
domain	The domain that contains the instance to resolve.
class	The class that contains the instance to resolve.
instance	The instance to resolve.
message	This specifies the message for conditional resolution paths.
default object	The default object. MVS only.

MVS Example

```
/*-----------------------------------------*/

 /* RESOLVE SYSTEMX.USER.  ZMASTER.ZUSERID    */

 /*-----------------------------------------*/

 DOMAIN='SYSTEMX ';

   CLASS ='USER    ';

        INST  = ZUSER;               /* SET INSTANCE EQ ZUSERID */

        INST  = LEFT(INST,32,' ');  /* PAD WITH BLANKS TO RIGHT */

        PATH  ='EDMSETUP';
```

```
DEFOBJ='ZMASTER '; /* DEFAULT OBJECT - USUALLY INBOUND OBJECT*/

PARM  = DOMAIN || CLASS || SUBSTR(INST,1,32) || PATH || DEFOBJ;

ADDRESS EDMLINK ZSIMRESO PARM; /* RUN SIMULATION TO GET SVC'S*/
```

UNIX Example

```
/********* SIMULATE ************ REXX * ********* 12/13/94  ****/

SAY 'SIMULATE BEGINS '


/*-----------------------------------------*/

/* RESOLVE SYSTEMX.USER.  ZMASTER.ZUSERID    */

/*-----------------------------------------*/

DOMAIN='SYSTEMX ';

CLASS ='USER';

INST  = 'USER1';

PATH  ='EDMSETUP';

PARM  = DOMAIN || ' ' || CLASS || ' ' || INST || ' ' || PATH;

ADDRESS EDMLINK ZSIMRESO DOMAIN CLASS INST PATH
```

Possible MVS Return Codes

Return Code	Description
0	All return codes for this method are zero.

Possible Non-MVS Return Codes

Return Code	Description
0	The method was successful.
8	An error was detected, the method failed.

EDMMSIGN

Method Name EDMMSIGN

Description This method enables you to authenticate an EDM Client session against the EDM database. The password stored in the ZPWD variable in the specified object is compared to the password stored in the user's profile. If the passwords match, the session continues unless the message "PASSWORD INVALID" is sent back to the EDM Client. Passwords can be changed by specifying the new password in "ZNEWPWD" and the old password in "ZPWD."

Usage You can use this method to authenticate users when an external security system is not in use.

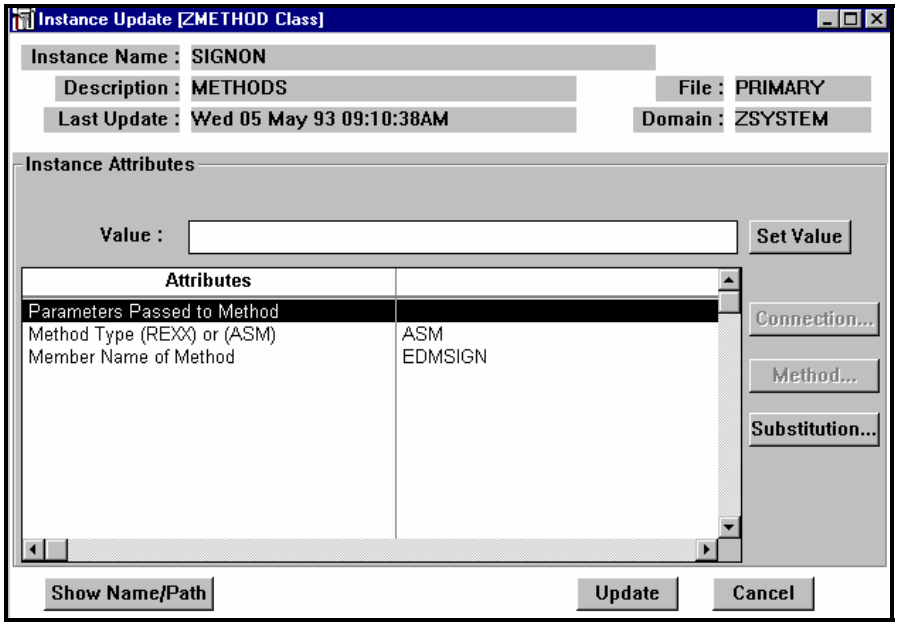
Type EDM Manager

Old Name EDMSIGN

Platform MVS, Solaris, HP-UX, AIX, OS/2, Windows NT.

Parameters Unix Optional: `object name, default object = ZMASTER`

Parameter Name	Description
object name	Specifies the name of the object from which the ZPWD variable is extracted.



Example

```
/*******************************REXX*****************************/
```

```
EDMMSIGN

     (Uses the ZMASTER Object)

EDMMSIGN & (ZCURRENT>ZCUROBJ)
```

Possible MVS Return Codes

Return Code	Description
0	The method was successful.
16	The password you have entered is invalid.
16	The password you have entered is missing.
16	A new password is required.
16	Duplicate userid count has been exceeded.

Possible Non-MVS Return Codes

Return Code	Description
0	The method was successful.
8	An error was detected, the method failed.

EDMMSINR

Method Name EDMMSINR

Description This method enables you to authenticate an EDM Client session against the password for the user found in an external security system such as RACF, Top Secret, or in Windows NT security. The password stored in the ZPWD variable in the specified object is compared to the password stored in the Unix database. If the passwords match, the session continues unless the message "PASSWORD INVALID" is sent back to the EDM Client. Note that passwords can be changed by specifying the new password in "ZNEWPWD" and the old password in "ZPWD" for MVS only.

Usage You can use this method to authenticate users when an external security system is in use.

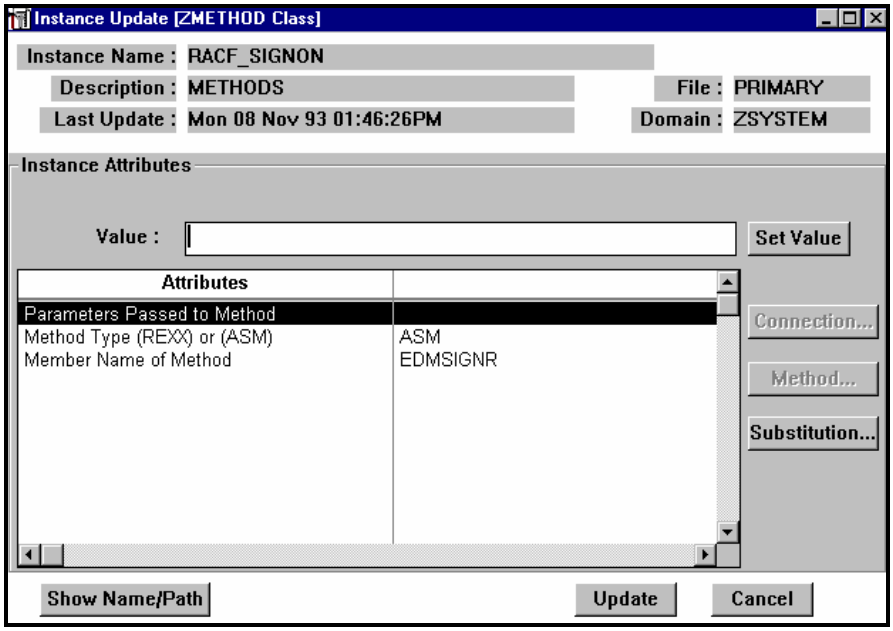
Type EDM Manager

Old Name EDMSIGNR

Platform MVS, Solaris, HP-UX, AIX, OS/2, Windows NT.

Parameters Unix Optional: `object name, default object = ZMASTER`

Parameter Name	Description
object name	Specifies the name of the object from which the ZPWD variable is extracted.



Example

```
/*******************************REXX*****************************/
```

```
EDMMSIGNR

     (Uses the ZMASTER Object)

EDMMSIGNR & (ZCURRENT>ZCUROBJ)
```

Possible MVS Return Codes

Return Code	Description
0	Method was successful.
16	An error was detected, the signon method failed.
16	Error, the variable could not be found in the object.
16	Duplicate userid count has been exceeded.

Possible Non-MVS Return Codes

Return Code	Description
0	The method was successful.
8	An error was detected, the method failed.

EDMMSORT

Method Name EDMMSORT

Description Sorts the instances of an in-storage object by the values of specified variables and according to the desired collating sequence.

Usage EDMMSORT can be used for reporting or testing purposes.

Type EDM Manager

Old Name ZOBJSORT

Platform MVS, Solaris, HP-UX, AIX, OS/2, Windows NT.

Parameters `SORT, object, variable1, variable2, variable3'`

Parameter Name	Description
sort sequence	Ascending (SORT) or descending (SORTD).
object	The name of the object to be sorted.
variable name(s)	Up to three variable names can be specified.

Example

```
/********* SIMULATE ************ REXX * ********* 12/13/94  ***/

SAY 'SIMULATE BEGINS '

SORTOBJ:

  PARM1 = 'SORT,'

  PARM2 = 'ZSERVICE'

  PARM3 = ',ZOBJNAME,ZOBJDATE,ZOBJTIME ';

  PARM  = PARM1 || PARM2 || PARM3;

  ADDRESS EDMLINK EDMSORT PARM;
```

Possible MVS Return Codes

Return Code	Description
0	The sort was completed.
4	The sort was not completed, less than two heaps.
16	Error, the object could not be found.
16	Error, the variable could not be found.
16	Storage is unavailable.
16	Error, $ REMQ failed.
16	Error, $ PUTQ failed.
28	Error, invalid parameters.

Possible Non-MVS Return Codes

Return Code	Description
0	The method was successful.
8	An error was detected, the method failed.

EDMMTUCH

Method Name EDMMTUCH

Description Updates the date/time stamp of an instance.

Usage You can use EDMMTUCH for testing and identifying problems occurring during object resolution.

Type EDM Manager

Old Name ZTOUCH

Platform MVS, Solaris, HP-UX, AIX, OS/2, Windows NT.

Parameters `domain(MAX_DOMAIN_LEN), class(MAX_CLASS_LEN), instance(MAX_INSTANCE_LEN)`

Parameter Name	Description
domain	The name of the domain that contains the instance to be updated.
class	The name of the class that contains the instance to be updated.
instance	The name of the instance to be updated.

Example

```
DOMAIN = "SYSTMEX";

CLASS  = "ZRSOURCE";

INST   = "TEST_OBJECT";

PARM= SUBSTR(DOMAIN,1,8)||SUBSTR(CLASS,1,8)||SUBSTR(INST,1,32);

ADDRESS EDMLINK EDMMTUCH PARM ;
```

Possible MVS Return Codes

Return Code	Description
0	Method was successful.
8	Error, no input parameters.
8	An error was detected, ZVCB is unavailable.
8	Object database access error.

Possible Non-MVS Return Codes

Return Code	Description
0	The method was successful.
8	An error was detected, the method failed.

EDMMULOG

Method Name EDMMULOG

Description Writes a message returned from the execution of a REXX method to a user log file. For this method to work, a new section will have to be added to the EDMPROF.DAT file.

Usage You can use EDMMULOG to write messages to a user log file.

Type EDM Manager

Old Name EDMMULOG

Platform Windows NT.

Parameters MSG

Parameter Name	Description
MSG	The message that will be written to the user log file. This message is returned by a method after its execution.

Example

```
ADDRESSS EDMLINK "EDMMULOG" MSG
```

Possible MVS Return Codes

Return Code	Description
0	Method was successful.
8	Error, no input parameters.
8	An error was detected, ZVCB is unavailable.
8	Object database access error.

Possible Non-MVS Return Codes

Return Code	Description
0	The method was successful.
8	An error was detected, the method failed.

EDMPROF.DAT Section

```
 [MGR_USERLOG]

ACTIVATE   = YES

DIRECTORY  = .

THRESHHOLD = 500000

FLUSH_SIZE = 128

COLUMN_WIDTH = 128

PIPE_SIZE  = 100000
```

EDMMUPSZ

Method Name EDMMUPSZ

Description Reads the ZRSOURCE object's size, and compressed size, and compares them to instances in the resource database.

Usage You can use this method to determine if proper resources were distributed as the result of object resolution.

Type EDM Manager

Old Name ZUPDSZ

Platform MVS, Solaris, HP-UX, AIX, OS/2, Windows NT.

Parameters `domain, option`

Parameter Name	Description
domain	
option	

Possible MVS Return Codes

Return Code	Description
0	Method was successful.
8	Error, no input parameters.
8	Error, no ZCMPSIZE available.
8	Error, no ZCMPSIZE field in class.
8	Error, ZCMPSIZE is not 8 characters in length.
8	Error, ZVCB is unavailable.
8	Object database access error.

Possible Non-MVS Return Codes

Return Code	Description
0	The method was successful.
8	An error was detected, the method failed.

EDMMVDEL

Method Name EDMMVDEL

Description Deletes all in-storage objects.

Usage You can use EDMMVDEL to refresh the EDM Manager after object resolution.

Type EDM Manager

Old Name ZVARDEL

Platform MVS, Solaris, HP-UX, AIX, OS/2, Windows NT.

Parameters No parameters allowed for EDMMVDEL.

Possible MVS Return Codes

Return Code	Description
0	Method was successful.
4	An error was detected, there are no objects.
16	Length error on FREEMAIN.
16	ZPCB queue empty or cannot be found.

Possible Non-MVS Return Codes

Return Code	Description
0	The method was successful.
8	An error was detected, the method failed.

EDMMVGBL

Method Name EDMMVGBL

Description Migrates values from one in-storage object to another. This method then deletes the source object.

Usage You can use EDMMVGBL to change object names after testing.

Type EDM Manager

Old Name ZVARGBL

Platform MVS, Solaris, HP-UX, AIX, OS/2, Windows NT.

Parameters `source, destination`

Parameter Name	Description
source	Object with values being migrated from
destination	Object with values being migrated to

Possible MVS Return Codes

Return Code	Description
0	Method was successful.
16	Error, object not found.
16	Error, instance not found.
16	Error, cleanup object not found.

Possible Non-MVS Return Codes

Return Code	Description
0	The method was successful.
8	An error was detected, the method failed.

EDMMXREF

Method Name EDMMXREF

Description EDMMXREF cross references class and instance usage during object resolution and collects information on the cross reference objects.

The EDMMXREF method will generate objects that enable administrators to crossreference users with any and all Departments, Workgroups, and Services that they are affiliated (connected) with.

➢ **To implement EDMMXREF method:**

1. Add new method instances to ZMETHOD. Some are methods to create the objects from the Primary database and others to write these objects to the Profile database.
2. Update the EDM Manager process (ZMASTER) used during the client connect process by adding new methods to ZSYSTEM.ZPROCESS.ZMASTER.
3. Update your class template(s), if necessary, to add new method attributes.
4. Update the _BASE_INSTANCE_ to specify the method instances to be executed. These items were created in step **1**.

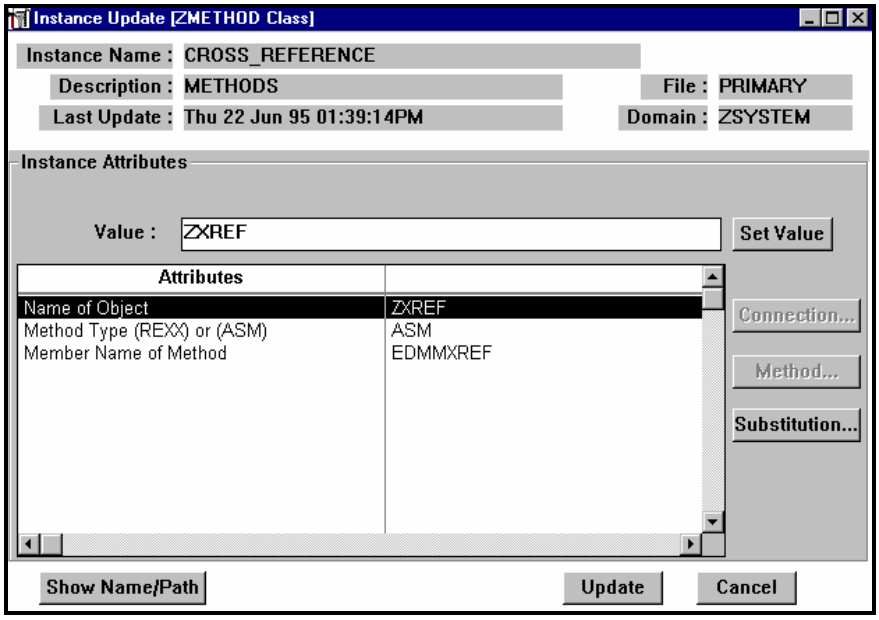
Usage You can use this method to collect and collate information on the results of object resolution.

Type EDM Manager

Old Name ZXREF

Platform MVS, Solaris, HP-UX, AIX, OS/2, Windows NT.

Parameters `object`



Possible MVS Return Codes

Return Code	Description
0	Method was successful.
8	An error was detected, method failed.

Possible Non-MVS Return Codes

Return Code	Description
0	The method was successful.
8	An error was detected, the method failed.

B Exporting Data Using the EDMMSQLP Method

This chapter provides you with the information you need to export data from the EDM Manager to an ODBC-compliant SQL database using the EDMMSQLP method.

Overview

The EDMMSQLP method provides EDM users with a tool to extract data from EDM's object-oriented database, and store it, via an ODBC connection, in a customer-specified SQL database table. Once data has been transferred to the SQL database, you can use the reporting capabilities of the SQL database system or third party reporting products that work with that system, to prepare reports on EDM's configuration, performance, activity and/or status.

EDMMSQLP is an EDM Manager method. By establishing a connection to this method in the distribution model for one or more desktops under management, administrators can cause the EDMMSQLP method to be invoked during the Client Connect Process of each such desktop.

Each invocation of EDMMSQLP causes one or more variables from a single EDM object to be transferred to the back-end database. If the EDM object consists of a single heap, the variables transferred will be stored in a single row of the back-end database table. If the EDM object consists of multiple heaps, variables from each heap will be transferred, one heap at a time, to individual rows of the back-end database table, one row per heap.

In a sense, EDMMSQLP provides users with a capability analogous to the EDM Profile file, implemented in an ODBC-compliant back-end database. The EDM Profile file stores EDM objects on a per-desktop basis, saved when the EDMMPPRO method is invoked during the Client Connect Process. By mirroring the use of EDMMPPRO with EDMMSQLP, you can save EDM data in an external ODBC-compliant SQL database, and report on EDM data using SQL-based end-user reporting capabilities.

Because EDMMSQLP is a generic tool for transferring data from an EDM object to a back-end database, you can make creative use of its capabilities to fit, given your organization's unique requirements.

For example, you may develop your own REXX method to package the data you need, from multiple EDM objects, into a single, multi-heap EDM object of your own design. This object would then contain only those variables needed for a certain report, gathered from the EDM objects pertaining to a set of users (or other entities) you need to be the subject of that report. You could then add a user to the EDM database whose sole purpose would be to build the custom object during its Client Connect Process and transfer the object's data to the back-end database. In effect, you would be constructing an EDM user whose Client Connect Process would mimic an SQL SELECT statement to extract a set of data from the EDM database, and to store it in the back-end database.

To make effective use of the EDMMSQLP method; you must know which variables you need to extract. Refer to the *EDM Class Reference* for this information.

Prerequisites

Configuring an ODBC Data Source

Before you can utilize EDMMSQLP, you must configure an ODBC *data source*. While it may seem confusing to call a place where extracted data will be stored a "data source"; this is ODBC terminology generically identifying a connection to an ODBC-compliant database.

ODBC is an API (**A**pplication **P**rogram **I**nterface) specification that allows SQL (**S**tructured **Q**uery **L**anguage) statements to be submitted from programs external to the database system, and processed by the database system over the ODBC connection. A database system exposes its ODBC interface to external programs through ODBC data source definitions.

EDMMSQLP is an external client program to the back-end database. The back-end database acts as a server.

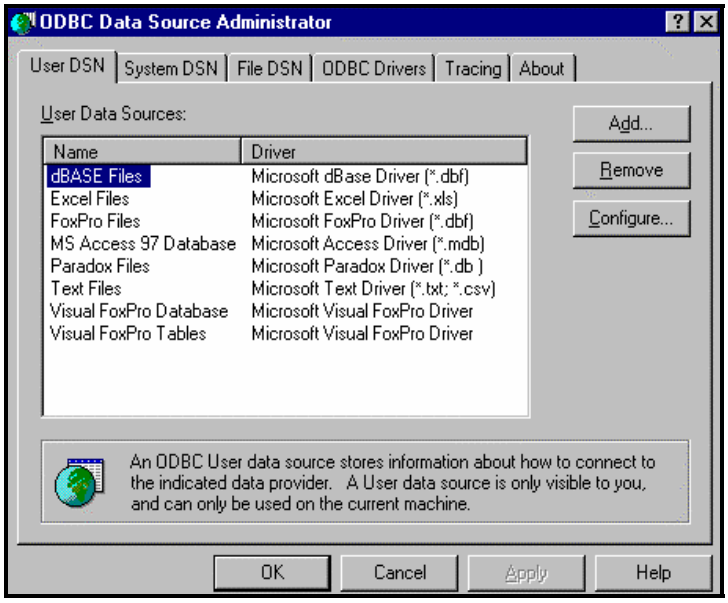
The ODBC data source definition identifies the back-end database connection and location where EDMMSQLP will store the data extracted from the EDM Database, and specifies any applicable options for how the back-end database system services the ODBC connection. Any such options will vary from back-end database to back-end database, and familiarity with the back-end database is needed to set them properly.

Many database systems are ODBC compliant, and the specifics of configuring an ODBC data source for a particular database system are described in that database system's documentation.

To configure an ODBC data source for a particular database system, the ODBC driver for the desired database system must first be installed on the computer on which the ODBC data source will be defined. This driver normally ships with the database system itself, or may be obtained separately from the database system vendor. There are also third party ODBC drivers available for most popular database systems.

The following example illustrates configuring the ODBC data source used in the examples in this chapter. The back-end database system is Microsoft Visual FoxPro.

1. First, open the Control Panel folder (click on the **Start** button, **Settings**..., **Control Panel**), and launch the ODBC Data Source Administrator accessed via the ODBC icon. A dialog similar to the following will appear.



IMPORTANT: If the EDM Manager is running as an NT service (which is more than likely), you must configure a System DSN (Data Source Name) rather than a User DSN. Click on the **System DSN** tab to open the System Data Sources panel.

2 Click on the **Add...** button to configure a new ODBC data source. Click on the driver for the database you intend to use, and click the **Finish** button. In our example, we click on the Microsoft Visual FoxPro Driver.

- The **Data Source Name:** is something you make up, and identifies this ODBC data source to external programs. You will supply this Data Source Name as a parameter to EDMMSQLP, as described later in this chapter.
- The Description field is free text. Make up a description that denotes the purpose or use of the data source definition.
- The Database Type setting pertains only to Visual FoxPro; you would not see this in dialogs to configure ODBC data sources for other back-end databases. This is also true of the Driver settings at the bottom of the dialog. These appear in the dialog when you click on the **Options** button; when you first open this dialog, the **Options** settings are hidden and the **Options** button is enabled.
- The Path specification identifies a folder where the ODBC-accessible Visual FoxPro data table(s) are located. You can enter the path to this folder via the keyboard, or press the **Browse...** button to open a dialog that enables you to navigate to and select the path from a list. Click on the **OK** button, and the ODBC data source definition is compete, and added to the system:

For many desktop-based ODBC-compliant database systems such as Visual FoxPro, Paradox, etc., configuring an ODBC data source is no more complicated than making up a name for the data source, specifying a path to the folder that contains the database table(s), and perhaps specifying a small number of database-specific settings.

ODBC data sources for server-based database systems such as Microsoft SQL Server; Oracle, Sybase, etc. are more complex to configure. This is typically the job of the system administrator for the database system in your organization. You will need to consult with your database system's administrator to have the ODBC data source you need configured properly in these cases.

Defining EDMMSQLP as an EDM Method

Before you can place a connection to the EDMMSQLP method in your distribution model, you must define at least one instance in the ZMETHOD class of the ZSYSTEM domain containing the information needed to execute the EDMMSQLP method. You can name the instance as you wish (in this example, it is named ZSQLPUT), consistent with the naming convention for EDM instances employed at your organization.

All of the variables in the instance should appear as shown here, except for ZMTHPRMS, the variable that contains or identifies the control information passed to the EDMMSQLP method when it executes. There are a number of different ways to pass control information to EDMMSQLP, and they are described in the next section of this chapter.

Passing Control Information to EDMMSQLP

EDMMSQLP requires a set of control information to perform its function. Control information is passed to EDMMSQLP as a parameter or set of parameters at execution time.

The control information may be passed to EDMMSQLP using one of the following ways.

1. As a parameter string passed on the command line (i.e. in the ZMTHPRMS variable of the PRIMARY.ZSYSTEM.ZMETHOD instance used to invoke EDMMSQLP). The maximum length of the parameter string on the command line is 255 characters. If the control information you need to pass to the method is longer than 255 characters, you must use one of the other options.

2. In a text file (identified by the CTRLFILE=<*file_name*> parameter).

3. In an EDM object (identified by the CTRLOBJ=<*object_name*> parameter)

   The following table identifies the control information required by EDMMSQLP.

Keyword	Description
CTRLFILE	The fully qualified name of a text file that contains the control information. If this parameter is present on the command line, the parsing of the parameter string stops and all the control information will be taken from the specified file. Required only if the control information is supplied in the specified file.
CTRLOBJ	The name of the EDM object that contains the control information. If this parameter is present in the command line, the parsing of the parameter string stops and all the control information will be taken from the specified object. Required only if the control information is supplied in the specified object.
SRCOBJ	The name of the EDM source object (right-padded with blanks to eight characters and enclosed in quotes, when specified in a text file or on the command line). The EDM source object contains the data to be transferred to the back-end SQL database.
SQLDSN	The ODBC data source name (DSN) to be used to connect to the back-end SQL database. See "*Configuring an ODBC Data Source"*, in this appendix, for details.
SQLTABLE	The fully qualified file name of the SQL table in the back-end SQL database into which EDMMSQLP will store the data.
SQLUSER	User ID to use in the database Connect process. Some back-end databases ignore this information; others require and verify it. See the database administrator of the back-end database in your organization for details.
SQLPASSW	The password to use in the database Connect process. Some back-end databases ignore this information; others require and verify it. See the database administrator of the back-end database in your organization for details.
SQLTOUT	Timeout value (in seconds) for the SQL Connect operation. If a successful connection to the back-end database cannot be established within this time, the connection attempt will be terminated by EDMMSQLP and an error logged in the EDM Manager log.
VC, or VC*nnn* where *nnn* is a sequential 3-digit number from 000 to the total number of variables to be transferred to the back-end database; used when control information is passed in an EDM object	Defines the correspondence between a variable in the EDM source object and the column in the back-end database table where it will be stored. One VC value must be specified for each variable-column pair participating in the operation. Specify: "VARNAME [,COLUMN_NAME][,U]" (include the quotes in a text file or on the command line) VARNAME is the name of the variable in the EDM object whose value will be transferred to the back-end database. COLUMN_NAME is the name of the column in the back-end database table that will receive the data. If COLUMN_NAME is omitted, VARNAME will be used; this assumes that the back-end database table's receiving column is named the same as the EDM object variable supplying its data. The third sub-parameter, [U] identifies the key field(s) used to locate the row in the back-end database to be replaced. At least one VC value must have the third sub-parameter coded.

Keyword	Description
PUTTYPE	Indicator for type of operation to be performed on the back-end database, either “R” for “Replace”, or “I” for “Insert”. When “Replace” is specified, EDMMSQLP will try to update the existing row first. If the row does not exist, EDMMSQLP will try to insert it. When “Insert” is specified, EDMMSQLP will try to insert the row. If this operation fails, no other action is taken. The default value is “R”.

Examples of the Parameter Strings

Example 1

```
CTRLFILE=D:\EDM\SQLTEST.TXT, SRCOBJ="ZCONFIG ", SQLDSN=EDM_Demo, SQLTABLE=EDM.DBF,
SQLUSER=ekahn, SQLPASSW=password, SQLTOUT=15, VC="ZHDWCOMP,DESKTOP,U", VC="ZHDWCPU,CPUTYPE",
VC="ZHDWMEM,MEMORY", VC="ZHDWOS,OS", VC="ZUSERID,USERNAME", PUTTYPE=R
```

In the example above, all the control information will be read from the file D:\EDM\SQLTEST.TXT. All other information specified in the parameter string, with exception of CTRLFILE itself, will be ignored.

Example 2

```
CTRLOBJ=SQLPARMS, SRCOBJ="ZCONFIG ", SQLDSN=EDM_Demo, SQLTABLE=EDM.DBF, SQLUSER=ekahn,
SQLPASSW=password, SQLTOUT=15, VC="ZHDWCOMP,DESKTOP,U", VC="ZHDWCPU,CPUTYPE",
VC="ZHDWMEM,MEMORY", VC="ZHDWOS,OS", VC="ZUSERID,USERNAME", PUTTYPE=R
```

In the example above, all the control information will be retrieved from the SQLPARMS object. All the other information from the command line will be ignored.

Example 3

```
CTRLOBJ=SQLPARMS,CTRLFILE=D:\EDM\SQLTEST.TXT
```

In the example above, all the control information will be retrieved from the D:\EDM\SQLTEST.TXT file; the information in the SQLPARMS object will be ignored. When both are specified, CRTLFILE takes precedence over CTRLOBJ.

Example 4

```
SRCOBJ="ZCONFIG ", SQLDSN=EDM_Demo, SQLTABLE=EDM.DBF, SQLUSER=ekahn, SQLPASSW=password,
SQLTOUT=15, VC="ZHDWCOMP,DESKTOP,U", VC="ZHDWCPU,CPUTYPE", VC="ZHDWMEM,MEMORY",
VC="ZHDWOS,OS", VC="ZUSERID,USERNAME", PUTTYPE=R
```

In this case, all the control information is received from parameter string in the command line.

Control information passed via text file

All control information may optionally be passed to EDMMSQLP in a text file. Use a text file to pass the control information when the total length of the parameter string exceeds 255 characters, or if you need to make multiple references to the same set of control information.

The format of the control information passed in a text file is exactly the same as the format of the parameter string passed on the command line. However, CTRLOBJ and CTRLFILE parameters will be ignored if found in the text file.

Control information passed via an EDM object

All control information may optionally be passed to EDMMSQLP in a control object. In this case, keywords described previously are specified as variable names in the EDM object. The repeating parameter, VC, must be specified in unique variable names. In order to create uniqueness, a three-digit index is appended to VC in forming the variable name. Thus, variable names will be VC000, VC001, VC*nnn*.

For instance, Example 4, may be implemented in a control object SQLPARMS, as follows:

Variable Name	Variable Value
SRCOBJ	ZCONFIG
SQLDSN	EDM_Demo
SQLTABLE	EDM.DBF
SQLUSER	ekahn
SQLPASSW	password
SQLTOUT	15
VC000	ZHDWCOMP,DESKTOP,U
VC001	ZHDWCPU,CPUTYPE
VC002	ZHDWMEM,MEMORY
VC003	ZHDWOS,OS
VC004	ZUSERID,USERNAME,U
PUTTYPE	R

The only parameter passed to the EDMMSQLP method on the command line in this case would be CTRLOBJ=SQLPARMS.

> **Important**: The three digit index appended to VC in forming the variable name **must** start with **000** and subsequent variable names must be created from an index value greater than the previous one. No numbers may be skipped.

Source EDM Object (SRCOBJ parameter) considerations

Any EDM object may be used as a source. In the source object, EDMMSQLP expects to find all the variables as defined in VC keywords in the control information parameter string, or in VC*nnn* variables in the control object.

Then, for each heap of the source object, EDMMSQLP reads the values of the requested variables and writes them into the SQL table's columns defined in the corresponding VC pair. All other variables in the source object that are not defined in any of the VC pairs of the control information are ignored. Generally, all the requested variables from a source object heap will be put in one row of the SQL table. The next heap of the

source object will provide values for the next row of the SQL table, and the process will continue until all the heaps of the source object are processed.

How To Invoke EDMMSQLP

This section provides you with the information you will need to invoke EDMMSQLP, and provides some examples.

Be sure the EDMPROF.DAT file is in the same directory where you execute the EDMMSQLP method. EDMPROF.DAT is located in the bin directory of the EDM Manager for NT. Then look for the path following the keyword DBPATH, which specifies the drive/directory where the EDM Manager files are located. You need to edit EDMPROF.DAT if it is pointing to a database other than the one you want to access.

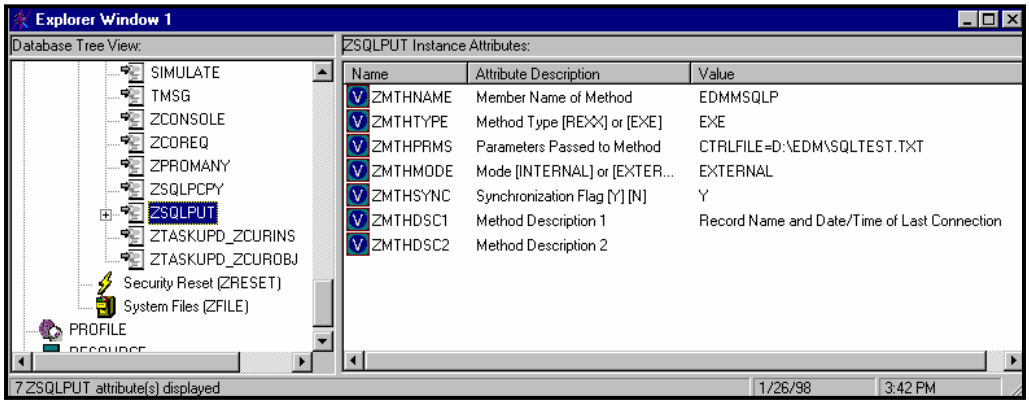
You may also want to make a backup copy of the EDMPROF.DAT file and the EDM Manager database as a routine precaution.

Review the information provided in the *"Prerequisites"* on creating an instance of PRIMARY.ZSYSTEM.ZMETHOD for EDMMSQLP. Before invoking the method, at least one such instance must be defined in the EDM Manager database. You may define multiple instances to invoke EDMMSQLP, where each instance (with its unique name) refers to a different set of control information in its ZMTHPRMS variable.

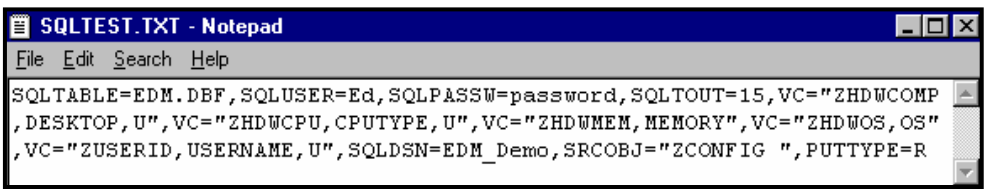
Example 5 – Mimicking the Profile File

In this example, we'll have the EDM Client Connect store information about the client desktop's hardware configuration in the backend database. The information will be extracted from the ZCONFIG object, and transferred to a backend Visual FoxPro table according to control information contained in a text file.

Here is the PRIMARY.ZSYSTEM.ZMETHOD instance created to invoke EDMMSQLP:



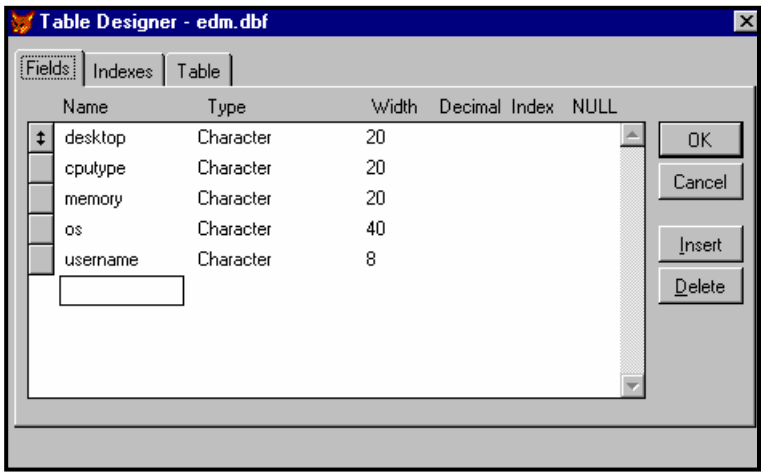
Here is the control information text file:



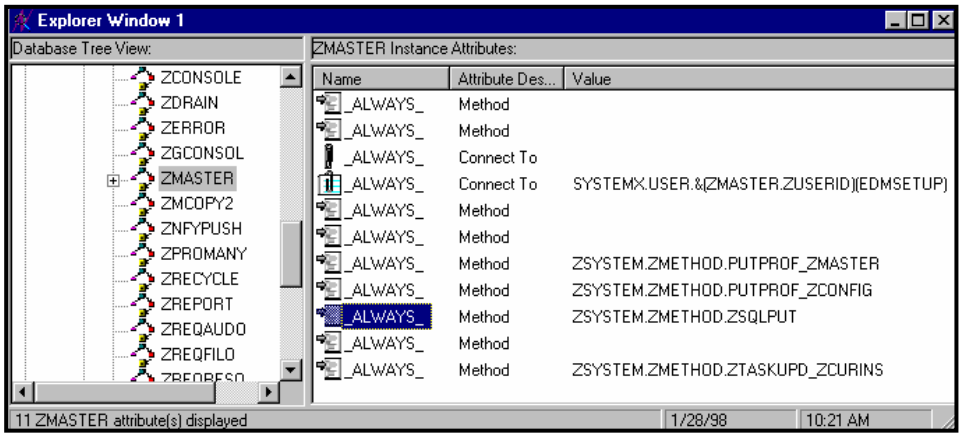
Note: The third sub-parameter "U" is coded on both the DESKTOP and USERNAME VC parameters, because the concatenation of these two fields is required in order to uniquely specify a client desktop in the back-end database.

The ODBC data source EDM_Demo is defined in the "*Configuring an ODBC Data Source*" section in this appendix.

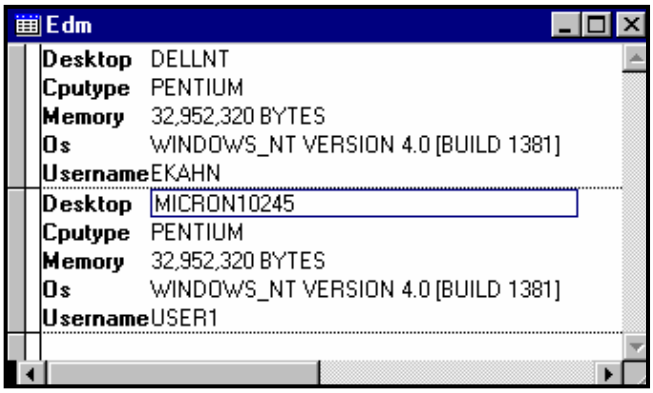
The structure of the Visual FoxPro EDM.DBF table is as follows:



To invoke the EDMMSQL method, a connection to the PRIMARY.ZSYSTEM.ZMETHOD instance named ZSQLPUT is added to the PRIMARY.ZSYSTEM.ZPROCESS ZMASTER instance, which is processed when the Client Connect sends the desktop ZMASTER object to the EDM Manager:



As each client connects to the EDM Manager, the pertinent fields are extracted from the client's ZCONFIG object, and stored in the Visual FoxPro EDM.DBF table:



Note: See how this example mimics the saving of the ZCONFIG object in the Profile file, which is accomplished via a connection to ZSYSTEM.ZMETHOD.PUTPROF_ZCONFIG in the ZMASTER instance, above.

Example 6 – Extracting from multiple objects

By design, EDMMSQLP extracts data from a single EDM object each time it is invoked. If you need to extract data from multiple objects, you must either invoke EDMMSQLP multiple times (once for each object), or write a custom EDM Manager method to compile data from multiple objects into a single object prior to invoking EDMMSQLP. This example demonstrates the latter method with a custom REXX method named SQLPHDW:

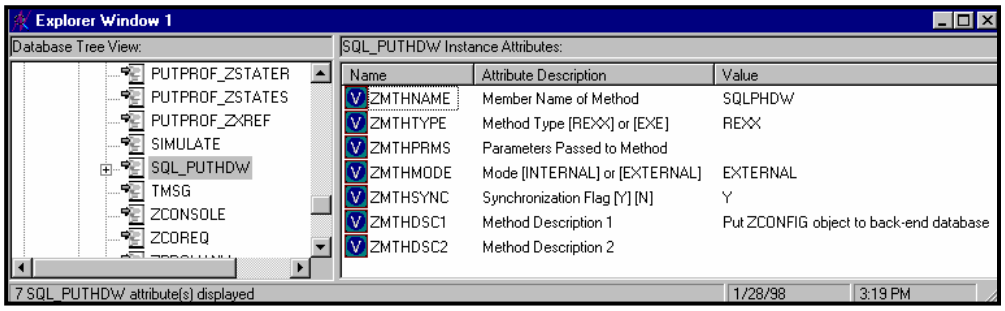
SQLPHDW REXX Manager Method

```
/*********************Put     SQL*********************/
/* will format and PUT a ZCONFIG object into an     */
/* ODBC compliant database                          */
/****************************************************/

/****************************************************/
/*                                                  */
/* COPYRIGHT NOVADIGM INC                           */
/* LICENSED MATERIAL PROPERTY OF NOVADIGM INC. 1994 */
/* NOVADIGM Enterprise Desktop Manager(tm)          */
/*                                                  */
/****************************************************/

/* ************************/
/* get the ZCONFIG object */
/* ************************/
address cmd
RC = EDMGET('ZCONFIG',1);                /* Get the ZCONFIG object  */
if RC <> 0 then exit
RC = EDMGET('ZMASTER',1);
ZCONFIG.ZOS = ZMASTER.ZOS || ' ' || ZCONFIG.ZHDWOSDB;
ZCONFIG.ZUSERID = ZMASTER.ZUSERID;
RC = EDMSET('ZCONFIG',1);

/* ***************** */
/* the main function */
/* ***************** */

RC = EDMGET('SQLCNTRL',1);
SQLCNTRL.PUTTYPE = 'R';
SQLCNTRL.SQLDSN = 'EDM_Demo';
SQLCNTRL.SQLTABLE = 'EDM.DBF';
SQLCNTRL.SQLTOUT = '10';
SQLCNTRL.SQLUSER = 'Ed';
SQLCNTRL.SQLPASSW = 'password';
SQLCNTRL.SRCOBJ = 'ZCONFIG';
SQLCNTRL.VC000 = 'ZHDWCOMP,DESKTOP,U';
SQLCNTRL.VC001 = 'ZHDWCPU,CPUTYPE';
SQLCNTRL.VC002 = 'ZHDWMEM,MEMORY';
SQLCNTRL.VC003 = 'ZOS,OS';
SQLCNTRL.VC004 = 'ZUSERID,USERNAME,U';
RC = EDMSET('SQLCNTRL',1);

params =  " CTRLOBJ=SQLCNTRL"
address edmlink edmmsqlp params
return;
```

This method combines variables from the ZMASTER object (ZOS, ZUSERID) with the variables in the ZCONFIG object. It then constructs an object (SQCNTRL) to contain the control information for a call to EDMMSQLP. It then invokes EDMMSQLP, passing the control object via the CTRLOBJ=SQLCNTRL parameter.

For further information on constructing custom EDM methods in REXX programming language, see *The EDM REXX Programming Guide*.

To invoke the SQLPHDW method, you must create an instance of the PRIMARY.ZSYSTEM.ZMETHOD class, which in this example looks like this:

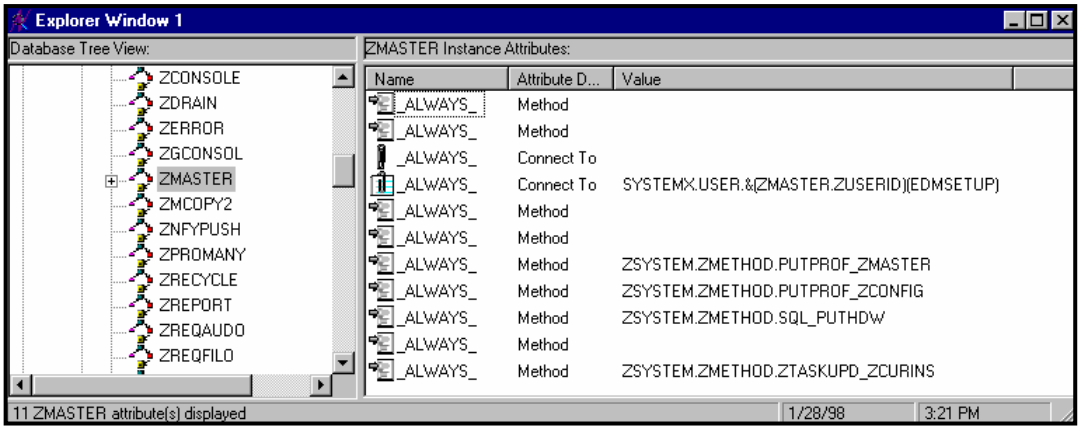


> **Note**: See that we've named the instance SQL_PUTHDW, and that the ZMTHPRMS variable is empty. The control information for EDMMSQLP is constructed internally within the SQLPHDW method and will be passed to the EDMMSQLP method in an EDM object.

The SQLPHDW file (without a file extension) must be located in the Manager\Rexx directory.

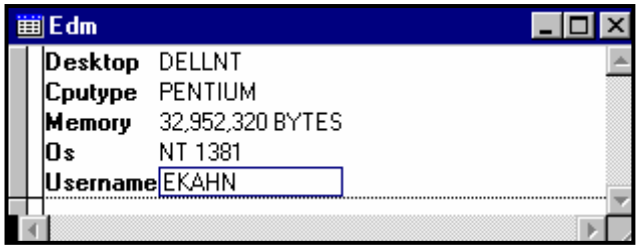
The ZMTHTYPE variable must be REXX for methods written in the REXX programming language.

To have the Client Connect invoke the SQLPHDW method, the ZSYSTEM.ZPROCESS.ZMASTER instance has been changed:



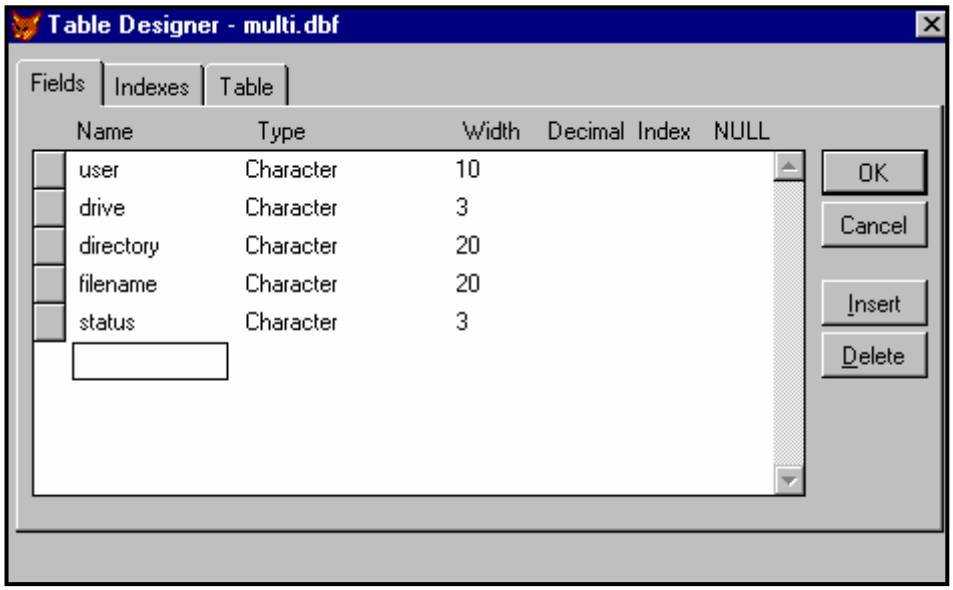
It now includes a connection to ZSYSTEM.ZMETHOD.SQL_PUTHDW.

As a result of the Client Connect , the data is transferred for the connecting desktop to the back-end database:



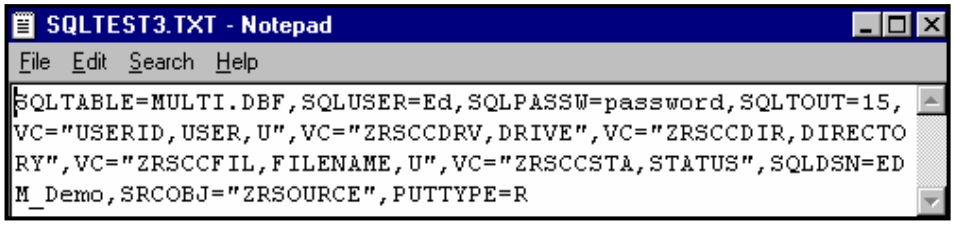
Example 7 – Transferring data from a multiple heap object

In this example, data is transferred from the ZRSOURCE object. The ZRSOURCE object has a heap for each resource (i.e. file) transferred to the client desktop.

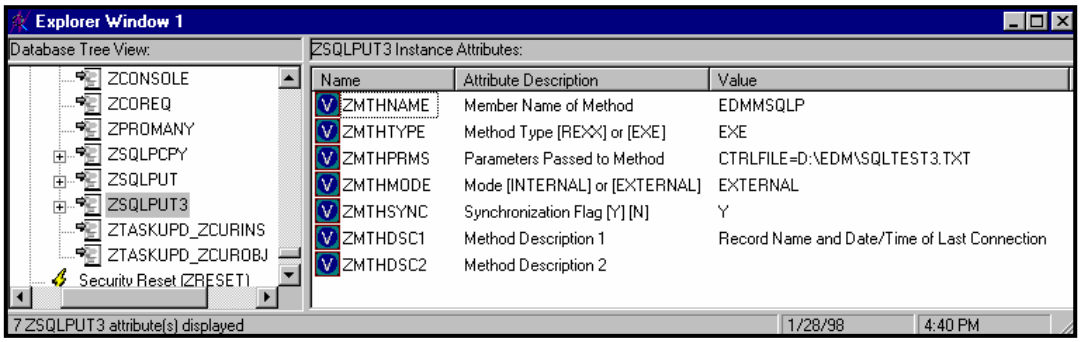


First, create a Visual FoxPro table to receive the data:

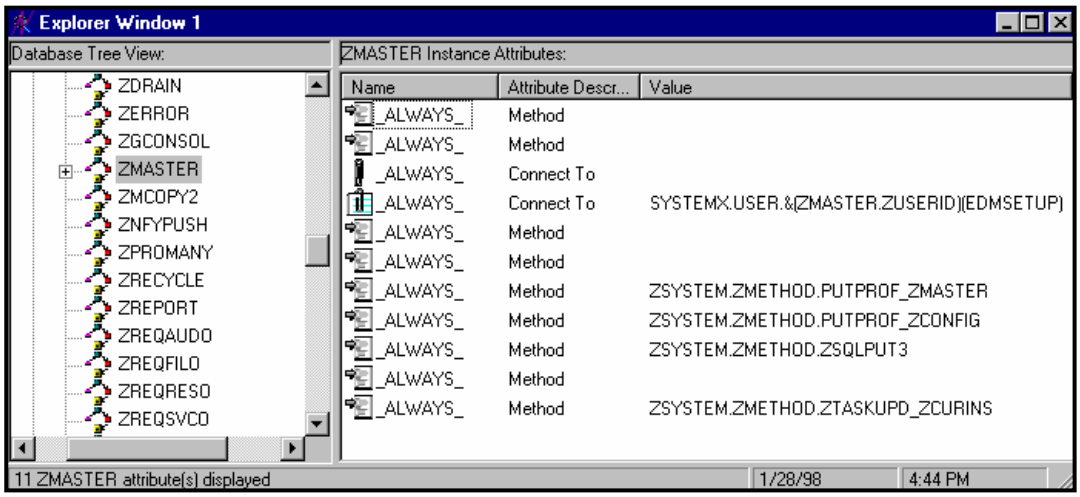
Second, create SQLTEST3.TXT, a control file for EDMMSQLP:



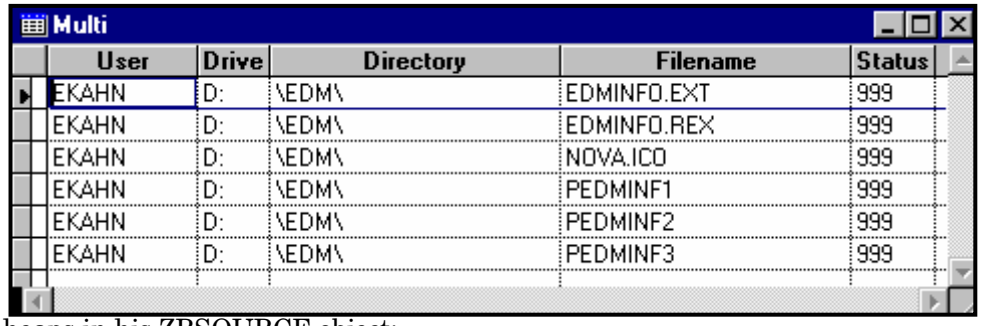
Third, create a ZSYSTEM.ZMETHOD instance named ZSQLPUT3 to invoke the EDMMSQLP method with the SQLTEST3.TXT control file:



Fourth, modify ZSYSTEM.ZPROCESS.ZMASTER to connect to ZSYSTEM.ZMETHOD.ZSQLPUT3:



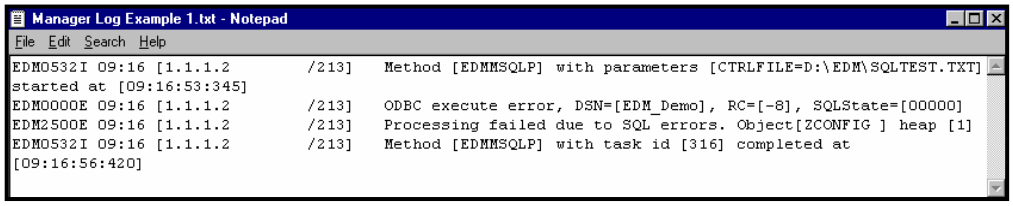
Finally, run the Client Connect . Here is the result in the Visual FoxPro table for one user, who had six heaps in his ZRSOURCE object:



Troubleshooting

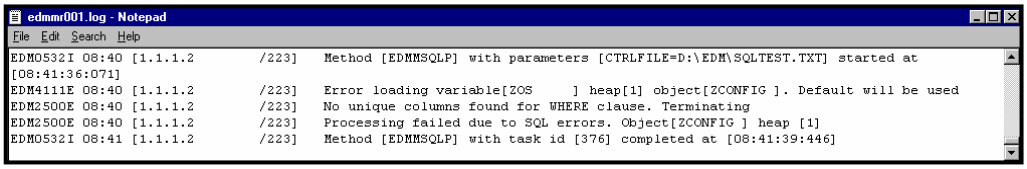
If data does not show up in the back-end database as expected, the first place to look when troubleshooting is the EDM Manager log.

Here is an example from the log:



In this example, the problem was that the back-end data table was opened exclusively by another application when EDM tried to update the table. You must take shared use of back-end tables into account when designing your data transfer strategy.

Here is another example from the log:



In this case, none of the VC pairs in the control information had the third sub-parameter ("U") coded, so EDMMSQLP was unable to identify a key field to use in a WHERE clause, which specifies which record in the back-end data table to update.

In general, the data extraction process is sensitive to typing errors. Check all typing carefully, and trace through all connections. Most of the time, your problem will be typographical.

Index

F

H

I

L

M

N

O

P

R

S

T

U

V

Z