



## **Administrator's Guide**

### **EsiObjects V4.2**

(c) Copyright 1994 - 2004, ESI Technology Corp, Bolton MA

This document contains the intellectual property of its copyright holder(s) and is made available under a license. If you are not familiar with the terms of the license, please refer to the license.txt file that is a part of the distribution kit.

Information in this document is subject to change without notice. Companies, names and data used in examples herein are fictitious unless otherwise noted. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of ESI Technology Corporation.

### ***Trademarks***

EsiObjects is a registered trademark of ESI Technology Corporation.

GT.M is a registered trademark of Sanchez Inc.

DSM, Cache, MSM are registered trademarks of InterSystems Corporation.

Microsoft, Visual Basic, Windows and Windows NT are registered trademarks of Microsoft Corporation.

# Table of Contents

<b>ADMINISTRATOR'S GUIDE</b> .....	<b>I</b>
<b>ESIOBJECTS V4.2</b> .....	<b>I</b>
<b>TABLE OF CONTENTS</b> .....	<b>III</b>
<b>INTRODUCTION</b> .....	<b>I</b>
<b>DOCUMENT CONVENTIONS</b> .....	<b>I</b>
<b>DEVELOPMENT CLIENT SERVER CONFIGURATIONS</b> .....	<b>2</b>
<b>STARTING UP THE DEVELOPMENT ENVIRONMENT</b> .....	<b>4</b>
STARTUP COMMAND LINE .....	4
STARTING THE ESIOBJECTS ENVIRONMENT .....	7
<b>CREATING A SESSION CONNECTION</b> .....	<b>9</b>
<b>SHUTTING ESIOBJECTS DOWN</b> .....	<b>10</b>
<b>MANAGING THE ESIOBJECTS LISTENER</b> .....	<b>11</b>
OVERVIEW .....	11
MULTIPLE CONNECTIONS TO THE SERVER .....	12
PLATFORM .....	12
MONITORING THE SERVER PROCESSES - ^VESOTCMN.....	12
AUTOMATING THE LISTENER STARTUP .....	14
<b>MANAGING SECURITY</b> .....	<b>16</b>
OVERVIEW .....	16
CREATING AND MAINTAINING USERS .....	16
MANAGING USERS FROM THE DEVELOPER'S UI.....	16

# Introduction

This guide contains information needed to administer the EsiObjects system on the supported M systems. It contains information about:

- 1) The EsiObjects Development Client Server Configurations.
- 2) MSM's Remote Volume Group (RVG).

## Document Conventions

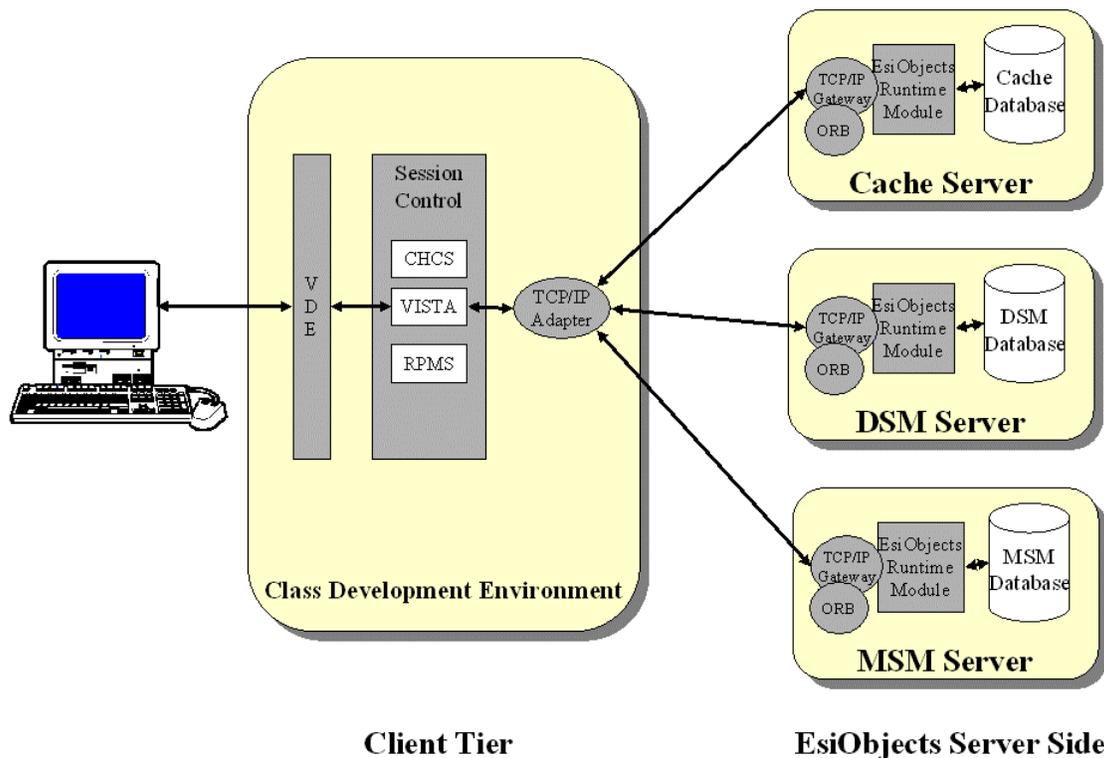
EsiObjects documentation uses the following typographical conventions:

For more information on this subject please refer to the <u>BREAK Command section of this guide.</u>	Underlined text is used to highlight a reference to another section of this guide or another guide.
<i>Property</i>	In text, italicized words indicate defined terms that are usually used for the first time. Words are also italicized for emphasis.
<b>CREATE</b>	Words in bold and capitalized are EsiObjects commands or keywords.
Set T%Test=I%Pat.Name	This font is used for code examples.

# Development Client Server Configurations

Illustrated below is the **EsiObjects Development Environment**. It is a combination of local components that reside on your development workstation and the EsiObjects Runtime that resides on your M server of choice.

The workstation contains all components that make up the **Class Development Environment (CDE)**. These components are dedicated to presenting all the objects that make up the contents of the EsiObjects system, that is, Libraries, Classes, Methods, Properties, etc.



Describing the CDE from left to right in the diagram above, the first component is the Presentation Layer. This component is responsible for presenting all meta-level objects to the developer via Microsoft's Window technology. All EsiObjects Browsers, Editor's and other tools are implemented using this technology.

The EsiObjects CDE is designed to offer you maximum flexibility. The Session Control component adds to that capability. It lets you set up any number of sessions to any supported M Server via the TCP/IP adapter. This adapter communicates with the TCP/IP gateway on the servers. The CDE presents you with a Session Control dialog that lets you define session connections. It also lets you activate sessions at your discretion. Multiple sessions can be active at any one time.

Connections between the EsiObjects Client and a Server are always made via the EsiObjects TCP/IP gateway. You can set up sessions that connect to a Cache, MSM, GT.M or DSM servers. For EsiObjects to run on any of these systems, you will be required to load the EsiObjects Runtime component. These M routines and globals are preloaded in volume sets (DSM), volume groups (MSM) or files (Cache and GT.M). The loading instructions are outlined on ESI's web site.

# Starting up the Development Environment

Activating the EsiObjects environment is accomplished by selecting the EsiObjects menu item or desktop icon. This executes the EsiObjects Command Line.

The **Startup Command Line** section describes the command line structure and qualifiers that are used to modify the startup of EsiObjects.

## Startup Command Line

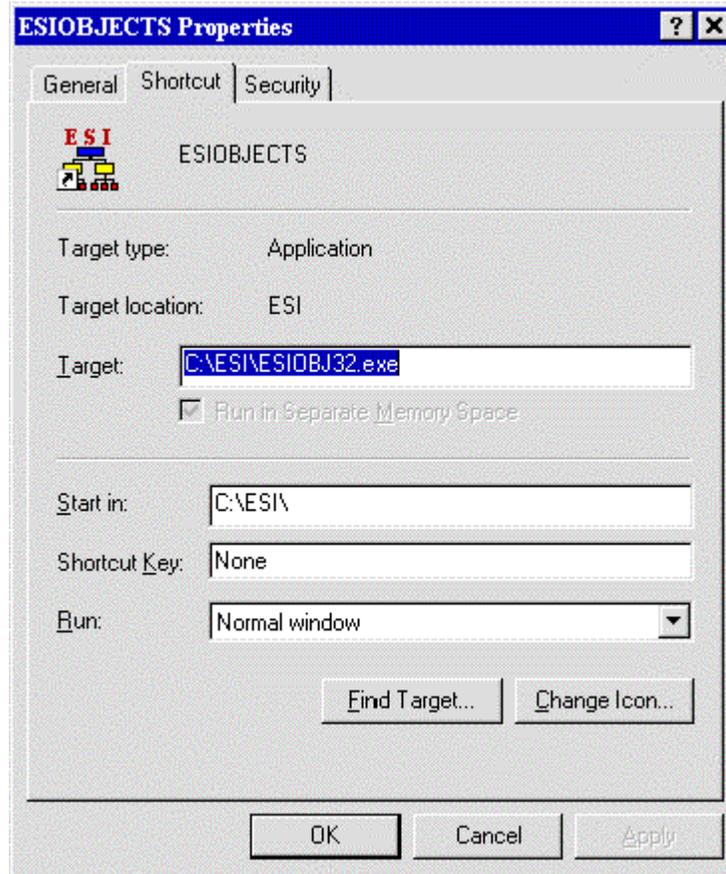
Generally, you will want to access EsiObjects directly from an Icon that is always on your desktop. The Icon has properties associated with it. They can be edited by accessing the property sheet for the icon. The Target line of the Shortcut sheet is where you customize the EsiObjects startup command line.

First, create a desktop shortcut to the Esiobj32.exe (or Esiobj32U.exe if you are on Windows NT or 2000) file that resides in the EsiObjects directory.

Next, right click on the EsiObjects icon, and then click on Properties.



Select the Shortcut tab sheet and alter the Target Field as desired.



### Command Line Structure

The general structure of the command line is as follows:

**ESI OBJ32.EXE**  $\Rightarrow$  [qualifiers]...

where:

$\Rightarrow$  is a space character.

**ESI OBJ32.EXE** can be a full path name.

**qualifiers** modify the meaning of the command line. Qualifiers begin with a forward slash (/) followed by the qualifier name.

**Note: Qualifiers must be separated with a space character otherwise the parser will not treat them as separate qualifiers and EsiObjects will not start up correctly.**

Specifically, the **qualifiers** affect the EsiObjects startup. Valid qualifiers are listed below.

<b><u>Qualifier</u></b>	<b><u>Description</u></b>
<b>/Admin</b>	<p>Brings EsiObjects up in Administrator mode. When you bring the Xecute Shell up in this mode, it will come up in the context of the \$Env object. This permits normal programming operations.</p> <p>Additionally, Administrative mode permits operations such as:</p> <ul style="list-style-type: none"> <li>• Checking the System check box on a Folder Property Sheet. When checked, normal users cannot alter all the properties of the folder.</li> <li>• Assigning Virtual ID numbers to virtual classes with in the range 1 – 1024.</li> </ul>
<b>/Debug</b>	<p>Starts up EsiObjects with internal debugging support enabled. Specifically, your menus will now include two new commands: <b>Inspect</b> and <b>Xecute</b>. The Inspect command will let you invoke the Object Browser on an EsiObjects structure. In other words, it lets you look under the hood of any object in the system, including those objects that define the EsiObjects system itself. Also, you can bring up the Xecute Shell under the same circumstances.</p> <hr/> <p><b>Note:</b> This switch should only be used when attempting to debug the EsiObjects system itself. It should never be activated when doing normal application work.</p> <hr/>
<b>/ESI</b>	<p>This qualifier enables all the features of /Admin and /Debug. However, when invoking the Xecute Shell, it will come up in the context of an object if that object is selected in the Session Browser. If not, it will come up in within the context of the \$Env object.</p>
<b>/NoConnect</b>	<p>Causes the client side to come up without automatically connecting to a server if AutoConnect is turned on.</p>
<b>/Update</b>	<p>Forces a database upgrade. Upgrades bring the EsiObjects database up to the current version level.</p>
<b>?</b>	<p>Brings up a message box informing the user of the valid command line qualifiers.</p>

### **Examples**

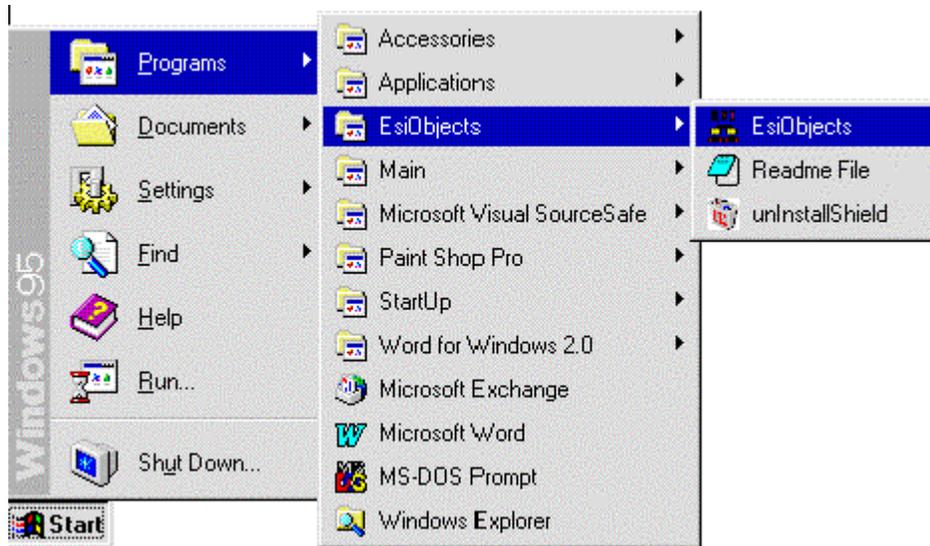
Note: The ESIOBJ32.EXE specified below will be ESIOBJ32U.EXE if you are on a Windows NT or 2000 system. The U denotes Unicode, no U denotes ASCII.

<u>Command Line</u>	<u>Description</u>
ESIOBJ32.EXE ?	Displays all qualifiers in a message box on your screen.
ESIOBJ32.EXE	Runs EsiObjects in normal mode where the Class Development Environment is brought up without any browsers.
ESIOBJ32.EXE /NoConnect	This qualifier overrides any automatic connections that the client side tries to make at startup. EsiObjects comes up without making server side connections.
ESIOBJ32.EXE /Debug	Brings EsiObjects up in Debug mode. EsiObjects will make two additional commands available to you via the menus: Inspect and Xecute. These commands let you browse and modify existing EsiObjects structures. This switch should be used for internal EsiObjects debugging only, not for normal application development.
ESIOBJ32.EXE /Admin	Brings EsiObjects up in Administrative mode. This allows the user to perform specific administrative level operations in EsiObjects such as controlling the properties of a folder.
ESIOBJ32.EXE /Update	This qualifier will force EsiObjects to run an upgrade on the session database that is connected. The upgrade operation will convert the existing database to the current version level.

## Starting the EsiObjects Environment

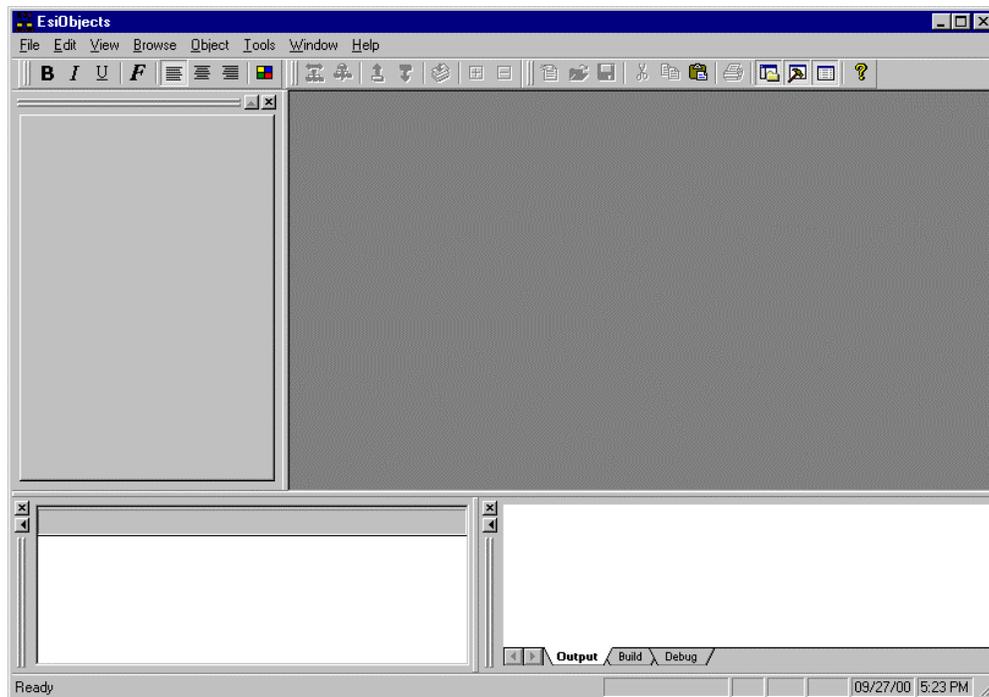
When you install EsiObjects, all supported menus are installed in the Start or Programs menu.

To start up EsiObjects click on the Start button, select the Programs option, then the EsiObjects path item and then the EsiObjects program item as shown in the diagram below.



If you created an icon on your desktop, double click on it to start up EsiObjects.

The Class Development Environment window appears as shown in the following figure.



You are now ready to begin working in EsiObjects. For more information about the Class Development Environment, see the [EsiObjects Programmer's Reference Guide](#) and the [EsiObjects Tools Guide](#).

## Creating a Session Connection

After EsiObjects has been installed you should create a session that will automatically connect to a database server at startup. If you are connecting to a Cache, DSM, GT.M or MSM Server, you must first install the EsiObjects Server files on the server of choice. You can download these files from the ESI web site. Each server file contains a readme file that explains how to set it up.

To connect to a MSM, DSM, GT.M or Cache database server, you will need to create a session that connects to the server via the TCP/IP Gateway.

Note that EsiObjects comes with one session defined called Local. It is set up to connect to a TCP/IP port 9000 on your local machine. If you are using an M system such as MSM Server or Cache on your local machine, you need only install the server side of the desired M server. Refer to the specific readme.txt file for the type of server you have installed locally.

If you are connecting to a remote server, please follow the steps below:

- 1) Select **File|Connections|Show** from the main menu bar. This will bring up a Session Control window.
- 2) Right-click inside the Session Control window, and select **New** from the popup menu. This creates a New Session dialog box.
- 3) Enter a name for the session in the Name field. Now select TCP/IP for the **Type**, and check the **Common Session** box. If you checked the **Auto connection** box, you will not have to select the session at startup time, the session will automatically connect. Click on the **OK** button to create your default session.
- 4) Right-click on the new session name in the Session Control window to bring up the popup menu. Then choose **Properties** from the popup menu. This displays a Properties form.
- 5) In the Connection tab sheet of the Properties dialog box, enter the IP address for the remote database server and the port on which it is running. Press the **OK** button to accept the new properties for the session.
- 6) Make sure the TCP/IP listener is running on the server. It must be running on the port you specified.
- 7) Now right-click on the session name and choose **Connect** from the popup menu. This connects you to the selected session. Make sure that the Session Browser is displayed. In that window you will see a tab sheet for the session appear. It will contain all the session's libraries and folders.
- 8) At this point you can start working with the EsiObjects Development Environment.
- 9) If you are using multiple database servers, you will need to define a separate session as above for each server that you connect to.

## Shutting EsiObjects Down

When you are done using EsiObjects, you can shut down the EsiObjects Environment.

Select **Exit** from the **File** menu to exit from EsiObjects. EsiObjects prompts you to save any unsaved changes you may have and closes all open windows. You may also shut down EsiObjects by clicking on the Close button in the upper right corner of the window.

Please note that this procedure shuts down the client, not the server.

# Managing the EsiObjects Listener

## Overview

There are a number of client connections that can be made into an EsiObjects server. Depending on the client, a specific component is used to connect to the server.

Client	Component
EsiObjects development workstation	TCP Bridge Dynamic Link Library (DLL)
VB/Delphi application	TCP Bridge ActiveX control
Web via Java Server Pages	
Web via Active Server Pages	TCP Bridge ActiveX control
Java application	Java Gateway DLL
C++ application	

In order to make the connection, each component listed above has properties that must be specified which indicate the TCP/IP address and port number to which this component will connect. On the server side, an EsiObjects *Listener* process must be started on the IP address and port number before the client can connect.

For example, on the server, the *Listener* process can be started up as a background job to accept client connections on port 8000.

```
>JOB START^VESOTCSV(8000)
```

The code shown above spawns off a background job that starts up a Listener process to accept a client connection on port 8000. This server process can service connections from any one of the clients listed above.

On the client, the example below shows the use of the TCP Bridge ActiveX control in a Visual Basic form. The *Address* and *Port* properties of the control must be set to the IP address and port number on which an EsiObjects server *Listener* will be running.

```
ESITCPBridge.Address = "123.456.789.10"
```

```
ESITCPBridge.Port = 8000
```

```
ESITCPBridge.Connect
```

If a Listener is running on port 8000 on the IP address 123.456.789.10, the `Connect` operation will connect the VB client to this Listener process. A Listener is dedicated to

the client that connects to it and services the requests for that client until the client disconnects the link.

## Multiple Connections to the Server

Multiple (simultaneous) clients can connect to port 8000. How that is accomplished depends on which Mumps (M) platform the EsiObject server is running.

Platform	M Line tag and Routine to accept multiple connections
DSM, MSM or GT.M	START^VESOTCPR( <i>port</i> , <i>start</i> , <i>end</i> )
Cache	START^VESOTCSV( <i>port</i> )

If the server is running in a Cache environment, multiple connections can be made to the *port* specified on routine invocation.

If running within DSM, MSM or GT.M, multiple connections can be made to the port specified in the first parameter to the VESOTCPR invocation. The process that gets started up in this case is called a *Redirector* because it accepts client connections and *redirects* the client to connect to a Listener running on an open port within the *start* and *end* range. The client, once redirected to the new port, begins operations with the Listener running on that port. The Redirector then waits for additional client connection requests to come in, which it will redirect to the next available port.

For example, on DSM, the command below spawns a background job that will start a Redirector on port 8000 and redirect client connections to any available port starting at 8001 up to port 8064.

```
>Job START^VESOTCPR(8000,8001,8064)
```

The first client connection that is made will be redirected to port 8001. The next connection will be directed to port 8002 and so on. Once port 8064 is reached, the Redirector will start over at port 8001 and look for the next open port. If all 64 ports are being used simultaneously, the Redirector will not allow further client connections until one of the ports becomes free – which occurs when a client disconnects from the server.

## Monitoring the Server Processes - ^VESOTCMN

A system manager or administrator can view the EsiObjects server processes that are currently running in the M environment by using the utility ^VESOTCMN. This program presents the user with a menu of operations that allow them to monitor and manage the EsiObjects processes.

The command below is entered at the M prompt within the area that EsiObjects is installed.

```
>DO ^VESOTCMN
```

The following menu is displayed on the screen.

EsiObjects TCP/IP SERVER MANAGEMENT MENU

1. Start TCP/IP Server
2. Shut Down TCP/IP Server
3. TCP/IP Server Status
4. Set Server Policies

Option :\_\_

This utility can be run on Cache or DSM environments.

#### Option 1: Startup

To start up a Listener that can accept multiple client connections, select option 1. You will be prompted to enter the port number on which to start the Listener and to which clients will connect. If this is being run in a DSM environment, you will additionally be prompted for a *Start* and *End* range of ports to which clients will be redirected. After specifying these parameters, the utility spawns a background job that opens the specified port and for client connections.

#### Option 2: Shutdown

You can shut down a Redirector or Listener process by selecting option 2. When you select this option, all currently running EsiObjects server processes are listed.

EsiObjects TCP/IP SERVER MANAGEMENT MENU

TCP/IP Server Status

The following ports are currently in use

Port	Start time	Type	Job #
8000	06/21/2001 3:39 PM	Redirector	320
8001	06/21/2001 3:39 PM	Listener	321

Shut Down server for which job number? :\_\_

The example above shows that a Redirector is running on port 8000, with a client Listener running on port 8001. The M job number is listed with each process. You shut down a specific process by entering in the M Job number.

#### Option 3: Status

To view the currently running EsiObjects server processes, select option 3. This lists all server Listeners/Redirectors running.

## EsiObjects TCP/IP SERVER MANAGEMENT MENU

## TCP/IP Server Status

The following ports are currently in use

Port	Start time	Type	Job #
8000	06/21/2001 3:39 PM	Redirector	320
8001	06/21/2001 3:39 PM	Listener	321

Enter 'R' to refresh or any key to exit>\_

As clients connect and disconnect the display above will show the current connections. Enter an "R" to refresh the display.

## Option 4: Policies

This option allows the system manager to specify parameters for the Listener processes that are created to service client connections. Currently there are two (2) policies that can be specified: (NOTE: any changes will not affect currently running listeners)

- *Listener timeout value (specified as minutes)*

This policy specifies the number of minutes of inactivity that will cause the listener to shutdown. The default is 1200.

- *Listener check frequency (specified as seconds)*

This policy determines how many seconds the listener should wait for input from the client before checking for session timeout or shutdown. You may enter a number between 5 and 120. The default value is 30.

## Automating the Listener Startup

Besides using the ^VESOTCMN utility (described above) to manually start any server Listeners, you may want to set up your system to automatically start up a Listener whenever the M environment is started. The method you use to accomplish this depends on your site capabilities and system management preferences. Below are given some possible alternatives.

### 1. For DSM, startup up from VMS

This option could be used to insure Listener startup whenever the VMS system restarts or after a backup, etc. The first task is to create a command procedure that will call into the M system to start up a Listener.

An example command file is below:

```
$ SET VERIFY
$ SET NOON
```

```
$!Start a redirector
```

```
$ DSM/ENV=dsm_env/VOL=vol/UCI=uci/DATA=port BATCH^VESOTCPR
```

Where: *dsm\_env* is the DSM environment

*vol* and *uci* are the DSM volume set and UCI in which EsiObjects server is installed.

*port* is the port number the Listener will open for client connections.

This command procedure will log into the specified DSM environment and invoke BATCH^VESOTCPR to start up a Redirector on the specified port. This assumes that the M environment has been started up and is accessible.

This command procedure can then be invoked from the VMS site-specific startup command procedure to insure the Redirector is started up upon system reboot. It could also be invoked from a backup command procedure after the system is restarted after a backup. In any case, you must make sure the M environment is started first, prior to having this command procedure invoked.

### *2. For Cache, upon system startup*

Cache provides for a procedure to be run automatically on system startup via their Configuration Manager and Cache Studio applications. Refer to the Cache documentation for setting up a Cache system to automatically run the START^VESOTCSV(*port*) routine when the system is started up.

### *3. For FileMan environments*

In the case where you want to insure automatic startup whenever the Fileman environment is restarted, you can modify Taskman to invoke the JOB^VESOTCPR(*port*) routine upon environment startup.

# Managing Security

## Overview

EsiObjects offers simple user name and password security. The usernames and passwords reside on the server. They can be created and maintained through the VESOTCMN routine or via the **Management|Security** command of the development UI.

## Creating and Maintaining Users

To set up users on the server using the VESOTCMN routine, you must have access to the MUMPS programming prompt within the ESI namespace that the VESOTCMN routine resides within.

Once you have logged into the area as a programmer, invoke the VESOTCMN routine.

```
> DO ^VESOTCMN
```

Select Option 5 from the menu.

At this point, you may enter a '?' in response to each prompt for information on how to answer.

## Managing Users from the Developer's UI

If the developer starts up the EsiObjects Development UI on a client PC with the /Admin (or /ESI) command line qualifiers, access to management functions are permitted through the Management main menu option. Refer to the [Class Development Environment Tools, Management Tools, Security](#) section of the [EsiObjects Tools Guide](#) for a complete description of these tools.