



Tivoli Business Systems Manager
Installation and Configuration Guide
Version 1.5



Tivoli Business Systems Manager
Installation and Configuration Guide
Version 1.5

Tivoli Business Systems Manager Installation and Configuration Guide, Version 1.5

Copyright Notice

© Copyright IBM Corporation 2001 All rights reserved. May only be used pursuant to a Tivoli Systems Software License Agreement, an IBM Software License Agreement, or Addendum for Tivoli Products to IBM Customer or License Agreement. No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise, without prior written permission of IBM Corporation. IBM Corporation grants you limited permission to make hardcopy or other reproductions of any machine-readable documentation for your own use, provided that each such reproduction shall carry the IBM Corporation copyright notice. No other rights under copyright are granted without prior written permission of IBM Corporation. The document is not intended for production and is furnished “as is” without warranty of any kind. **All warranties on this document are hereby disclaimed, including the warranties of merchantability and fitness for a particular purpose.**

U.S. Government Users Restricted Rights—Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corporation.

Trademarks

IBM, the IBM logo, Tivoli, the Tivoli logo, AIX, Cross-Site, NetView, and TME are trademarks or registered trademarks of International Business Machines Corporation or Tivoli Systems Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both. OMEGAMON is a registered trademark of the Candle Corporation.

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

Notices

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

Contents

Preface.....	ix
Packaging Contents	ix
Who Should Read This Guide	x
Prerequisite and Related Documents	x
What this Guide Contains.	xi
What Is New for TBSM Version 1.5	xii
Typeface Conventions	xiii
Contacting Customer support	xiv
 Chapter 1. Introducing TBSM	1
TBSM Architecture	2
Object State Management	3
Tivoli Source/390	3
SQL Database Server	4
TBSM Management Server	4
TBSM Workstation	4
Service Desk.	4
Availability Reporting System	4
TBSM - Distributed Architecture	5
Event Enablement	5
Task Server	5
Instrumentation	6
 Chapter 2. Installing TBSM Base Services	7
Language Support	7
Installing SQL Database Server	8
Installing SQL Server 7.0 Service Pack 2.	8
Changing the SQL Server Authentication Password	9
Turning Off Certain Client-Side Options	9
Installing TBSM Base Components	10
Installing TBSM Enterprise Edition	10
Installing TBSM Distributed Edition	19
 Chapter 3. Installing TBSM/D Components	25
Existing GEM Customers	25
Installing TBSM - Distributed Edition on the TEC Server	25
Installing Components from the Tivoli Desktop	26

Installing Components from the Command Line.	28
Configuring the Connection between TBSM and the Tivoli Enterprise Console Server	30
Installing Tivoli Instrumentation Service	31
Other Installation Options	32
Chapter 4. Configuring TBSM/D Edition	33
Configuring Event Enablement and Task Server.....	33
Setting Security for Running the Task Server and Event Enablement	33
Configuring TCP/IP Port Numbers.....	34
Configuring for Network Address Translation.....	35
Configuring Multiple Tivoli Object Servers on the TEC Machine.....	35
Configuring the Task Server	35
Configuring Event Enablement	38
Monitoring Tivoli Instrumentation Service	40
Configuring Support for a Non-Local TEC	41
Configuring for TMA Endpoints	41
Configuring for Managed Nodes	41
Configuring Support for Multiple TECs.....	41
Configuring Monitoring of Tivoli Instrumentation Service	43
Chapter 5. TBSM Enterprise Edition Components	45
TBSM Source/390.....	45
TBSM Source/390 Address Spaces	46
TBSM Source/390 Object Pump	46
TBSM Source/390 Object Pump Architecture.....	47
TBSM Source/390 Object Server.....	48
TBSM Source/390 Dataspace.....	48
Chapter 6. Installing & Configuring TBSM Source/390.....	49
VTAM Definitions.....	49
3270 Pool Terminals	50
LOGMODE Entries for 3270 Pool Terminals.....	51
Tivoli NetView Customization.....	51
NetView Started Task JCL.....	52
NetView DSIPARM Member Customization	52
MVS Performance.....	53
Running Multiple Copies of Source/390 on an MVS Image	53
JES3 Support	53
TBSM Startup Parameter Considerations	53

TBSM Source/390 Dataspace Startup Parameters	54
TBSM Source/390 Object Server Startup Parameters	55
LU6.2 Interface	55
TCP/IP Interface	55
Optional Parameters for TCP/IP Interface.	57
Common Parameters for Both LU6.2 and IP Interfaces.	58
Object Server Problem Analysis.	59
Connectivity Problems.	59
Event Tracing	59
Source/390 Security.	60
OPERCMDS Security Definition	60
Defining an OMVS Segment for TBSM Source/390 Object Pump	60
TBSM Source/390 Object Pump Startup Parameters.	62
Installing the CICS Transient Data Exit Program	67
Installing the Operations Planning and Control (OPC) EDI.	68
External Data Interface and Automation Products.	69
Data Sources.	69
Messages versus Exceptions versus State Change.	70
Functional Components	70
Application Capture.	71
REXX Programs	71
EDI.	71
Automation (Starting/Ending)	72
TBSM Source/390 Object Pump	73
Required Application Capture Modifications	73
REXX Program Call	75
Event Flow	75
Installing the System Automation for SA/390 EDI	76
Installing the OPS/MVS EDI.	77
Installing the AF/OPERATOR and Auto Operator EDI.	78
Required NT Changes for EDI.	78
Installing the TBSM Omegamon Interface	79
Requirements for Omegamon from the TBSM Workstation	79
TBSM Source/390 Object Server Modify Commands.	80
TBSM Source/390 Object Pump Modify Commands	81
TBSM Source/390 Object Pump Commands	84
Handling the Subsystem Max Threads Condition	86

Bulk Discovery	87
GTMAOPE0 Utility Configuration.	87
ASIMVSIListener TBSM Server Configuration	88
GTMAOPE0 Problem Analysis	89
Installing LU6.2 Support	90
Configuring VTAM environment using PU2.1 Node.	90
Chapter 7. Installing Host Integration Server	93
Host Integration Server Security	93
Installing the Host Integration Server	93
Configuring Host Integration Server 2000	94
Installing Microsoft Host Integration Client	99
Configuring the Host Integration Server Client.	99
Installing Event Handler Server (Configuration 1)	100
Installing Event Handler Server (Configuration 2)	101
Validating LU6.2 Support	101
Installing Event Handler Server Using IP (Configuration 3)	102
Adding Additional Operating Systems	102
Appendix A. Hardware	105
Differences between Enterprise Edition and Distributed Edition	105
NT Servers (Enterprise Edition)	105
NT Servers (Distributed Edition)	106
NT Workstation	106
Appendix B. Enterprise Edition Hardware & Software	107
Hardware	107
Software	107
Hardware and Software Requirements for Testing or Quality Assurance	109
Hardware	109
Software	109
Hardware and Software Requirements - Workstations	110
Hardware.	110
Software	110
Appendix C. Distributed Edition Hardware & Software	111
Hardware.	111
Software	111
TBSM - Distributed Edition Installed Components	112

Appendix D. Software Requirements for Non-Windows Platforms	113
Appendix E. Network Configuration Using Firewalls.....	115
Configuring TBSM Windows NT Components	116
Windows NT Registry Modifications	116
TBSM SQL Server Configuration	117
Appendix F. Layouts & States for Automation EDI.....	119
Application Capture Modification Samples	119
Record layout for State Changes, Messages, and Exceptions.	122
Available States for Automation EDI	124
Appendix G. SA/390,OPS/MVS, and AF/Oper EDI Variables	125
Appendix H. Installing/Configuring SNA Server.....	127
SNA Client/Server Security	127
Installing the SNA Server	127
Configuring the SNA Server	128
Installing SNA Client.	133
Configuring the SNA Client.	133
Index.....	135

Preface

This Tivoli Business Systems Manager Installation and Configuration Guide describes how to install and configure the Tivoli Business Systems Manager (TBSM) product. This Guide contains information for both the Tivoli Business Systems Manager Enterprise Edition and the Tivoli Business Systems Manager Distributed Edition.

The Tivoli Business Systems Manager Enterprise Edition includes Source/390, which is a component that resides on OS/390 and collects availability data from a variety of OS/390 based sources. Some of these sources are the OS/390 console, performance monitors, schedulers, started task managers, and other third party applications.

The Distributed Edition is the same product as the Enterprise Edition without the Source/390 component.

Much of the information found in this document applies to both Editions. However, when the information applies to only one edition, this is noted. Although the Enterprise Edition contains the components to manage the distributed environment as well as the OS/390 environment, many Enterprise Edition customers will choose to implement monitoring their OS/390 environment before implementing the monitoring of their distributed environment. Therefore, we differentiate the components that are used for monitoring the distributed environment and the base components that are used for monitoring both environments.

Packaging Contents

Tivoli Business Systems Manager is packaged on four CDs and one 3490 cartridge tape. The list of CDs and their contents are as follows:

- **Tivoli Business Systems Manager V1.5 Base Services CD** – contains all of the common services and components that are used for both the Enterprise Edition and the Distributed Edition. These services are Windows based and include the components to monitor the OS/390 environment and the new Java console.
- **Tivoli Business Systems Manager V1.5 Distributed Edition CD** – contains all of the services and components that enable monitoring and management of distributed systems and is necessary for integration of products such as Tivoli Manager for MQ Series and Tivoli Manager for Domino. (This is the Tivoli Desktop install image.)
- **Tivoli Business Systems Manager V1.5 Publications CD** – contains all of the product documentation in the PDF format. Documentation includes the Tivoli Business Systems Manager User's Guide, Administration Guide, and this Installation and Configuration Guide. It also includes any Release Notes and/or ReadMe files that contain updated information that did not make it into the guides before publication.
- **Tivoli Business Systems Manager V1.5 NLS Language Support CD** – contains translated PII in the following languages:

French	Japanese	Simplified Chinese
German	Korean	Spanish
Italian	Brazilian Portuguese	Traditional Chinese

The cartridge tape and its contents:

-
- **Tivoli Business Systems Manager V1.5 3490 Cartridge Tape** – contains the services and components that are OS/390 based and that are necessary to monitor that environment. This includes the monitors for the OS/390 operating system and its various subsystems such as CICS, DB2, IMS, OPC, and others.

Product	What this includes:
TBSM Distributed Edition	Base Services CD, Distributed Edition CD, and Documentation CD
TBSM Enterprise Edition	Base Services CD, Distributed Edition CD, Documentation CD, and the 3490 cartridge tape

Who Should Read This Guide

This guide is intended for network administrators and system programmers responsible for installing, configuring, and using TBSM.

Prerequisite and Related Documents

The following documents are shipped on the Publications CD with the TBSM product:

- *Tivoli Business Systems Manager Installation and Configuration Guide*
- *Tivoli Business Systems Manager Console User's Guide*
- *Tivoli Business Systems Manager Administration Guide*
- *Tivoli Business Systems Manager Distributed - Overview*
- *Tivoli Business Systems Manager TSD Integration/Configuration*
- *Tivoli Business Systems ManagerTSD OS/390 Integration/Configuration/Installation Guide*
- *Tivoli Business Systems Manager CA-7 Release Notes*
- *Tivoli Business Systems Manager RODM Release Notes*
- *Tivoli Business Systems Manager Messages Guide*
- *Tivoli Business Systems Manager SMS Release Notes*
- *Tivoli Business Systems Manager TMON (MVS, DB2, CICS/ESA) Release Notes*
- *Tivoli Business Systems Manager MainView (CICS, DB2, IMS, OS/390) Release Notes*
- *Tivoli Business Systems Manager WebSphere for OS/390 Release Notes*
- *Tivoli Business Systems Manager CICSplex SM Release Notes*
- *Tivoli Business Systems Manager DB2 and DB2 PM Release Notes*
- *Tivoli Business Systems Manager IMS Release Notes*
- *Tivoli Business Systems Manager Operation and Planning and Control Release Notes*
- *Tivoli Business Systems Manager RMF Release Notes*
- *Tivoli Business Systems Manager Systems Automation for OS/390 Release Notes*
- *Tivoli Business Systems Manager User's Guide*

Depending on your TBSM task, the following is a list of related publications:

-
- *Tivoli Manager for OS/390 Enterprise Business Systems Management (redbook)*
 - *Tivoli Business Systems Manager CICSplex SM Instrumentation Program Directory*
 - *Tivoli Business Systems Manager DB2 for OS/390 Instrumentation Program Directory*
 - *Tivoli Business Systems Manager Program Directory*
 - *Tivoli Business Systems Manager Program Directory Distributed Edition*
 - *Tivoli Business Systems Manager Tivoli Ready Enablement Program Directory*
 - *Tivoli Business Systems Manager Product Release Information*
 - *Tivoli Global Enterprise Manager CICSplex SM Instrumentation Program Directory*
 - *Tivoli Global Enterprise Manager DB2 for OS/390 Instrumentation Program Directory*
 - *Tivoli Global Enterprise Manager Tivoli Ready Enablement Program Directory*
 - *Tivoli Distributed Monitoring User's Guide*
 - *Tivoli Software Distribution User's Guide*
 - *Tivoli Task Library Language Developer's Guide*
 - *IBM TCP/IP User's Guide*
 - Tivoli Enterprise Console library
 - Tivoli Management Framework library
 - Tivoli NetView for OS/390 library
 - IBM CICSplex System Manager for MVS/ESA library
 - IBM DB2 PM for OS/390 library
 - IBM DB2 UDB for OS/390 library
 - The DB2 UDB for Windows NT (or OS/2 V6) on-line books

For instrumentation, familiarity with the Application Management Specification (AMS), an industry-endorsed specification for creating management-ready applications is required. A copy of the AMS can be found at: www.tivoli.com/products/index/module_designer/

What this Guide Contains

The Tivoli Business Systems Manager Installation and Configuration Guide covers the following topics:

Introducing TBSM

Introduces Tivoli Business Systems Manager, explains the Line of Business concept, and describes the architecture.

Installing TBSM Base Services

Includes instructions for installing SQL database server and for installing Base services and components.

Installing TBSM - Distributed Edition Components

Gives instructions for installing the Distributed Edition of Tivoli Business Systems Manager.

Configuring TBSM - Distributed Edition

Gives instructions for Configuring the Distributed Edition of Tivoli Business Systems Manager.

TBSM - Enterprise Edition Components

Describes the Source/390 Components that enable monitoring and management of an OS/390 based system.

Installing & Configuring TBSM Source/390

Gives instructions for Configuring the Enterprise Edition of Tivoli Business Systems Manager.

Installing and Configuring Host Integration Server

Describes how to install and configure Microsoft Host Integration Server.

Hardware

Provides information about hardware needed to support Tivoli Business Systems Manager.

Enterprise Edition Hardware and Software Requirements

Provides information about the hardware and software requirements needed to support the Enterprise Edition of Tivoli Business Systems Manager.

Distributed Edition Hardware and Software Requirements.

Provides information about the hardware and software requirements needed to support the Distributed Edition of Tivoli Business Systems Manager.

Network Configuration Using Firewalls

Provides information about communication between TBSM servers and its client environment through a firewall or router.

Layouts & States for Automation EDI

Provides application capture modification samples, record layouts, and available states for automation EDI.

Installing/Configuring SNA Server

Describes how to install and configure a Microsoft SNA Server.

What Is New for TBSM Version 1.5

New information includes:

- Support for Windows/2000 as a client or server.
- Support for TCP/IP for customers choosing not to use the SNA LU6.2 protocol. Formerly the connection between the Source/390 component (specifically the TBSM Object Pump) and the TBSM distributed servers required SNA and a Microsoft host server product. A new connection allows this communication pipe to be TCP/IP.
- Distributed support for Computer Associates (CA) The Next Generation (TNG). This support enables loading TNG objects from the TNG database to the TBSM database. Events and messages for TNG objects are captured and used to update the TBSM GUI with availability information for TNG resources.
- Support for feeding events from 3rd party monitoring products into the TBSM database with these objects. Events and messages for the objects are captured and used to update the TBSM GUI with information about the resources' availability status.
 - BMC MainView for CICS

-
- BMC MainView for DB2
 - BMC MainView for IMS
 - BMC MainView for MVS
 - Candle Omegamon for IMS
 - Landmark The Monitor (TMON) for CICS
 - Landmark TMON for DB2
 - Landmark TMON for MVS
- Support for Service Desks/Tivoli Service Desk for OS/390. The capability to manipulate problem management records (Add, Update, Delete) is now available for extension. The Tivoli Professional Services group, or similarly skilled internal services groups, will be able to use other problem management software (including Peregrine and Remedy) in addition to Tivoli's Service Desk products. A document that describes how to add a new problem management software package will be provided. An API is required from each problem management software product.
 - Utilities for assisting with the installation and deployment of TBSM
 - Support for IBM's WebSphere. This support enables the discovery of WebSphere on OS/390. This release provides an object view of WebSphere/390 and covers objects that are instantiated as MVS address spaces.
 - Support for NetView for z/OS. This support enables the discovery of APPN SNA objects using Resource Object Data Manager (RODM). Events and messages for the objects are captured and used to update the TBSM GUI with information about the resources' availability status.
 - Support for IBM's System Managed Storage (SMS). This support enables the discovery of storage objects. Events and messages for these objects are captured and used to update the TBSM GUI with information about the resources' availability status.
 - New graphical user interface (GUI) for TBSM that consists of a JAVA-based console. This support provides the capability to run a TBSM client on Sun, AIX, Linux, and the Windows Family. Additionally, the Java GUI provides:
 - Ability to launch products from the TBSM GUI. Launch points are defined for launching directly into NetView's NMC. Support is also included for enabling other applications to be launched from the TBSM GUI.
 - Security capabilities that use Windows-based security to validate Windows logons and passwords. Users log on to the client specifying their domain-qualified Windows login name and password. This information is encrypted and sent to the Application server, which attempts to authenticate the user and returns to the client an indication of either a login failure or the user's authorization.

Typeface Conventions

This document uses several typeface conventions for special terms and actions. These conventions have the following meanings:

- Bold** Commands, keywords, file names, authorization roles, flags, universal resource locators (URLs), or other information that you must use literally appear in bold. The names or titles of screen objects and application components also appear in bold.

Italics Variables and values that you must provide appear in italics. New terms appear in italics when they are defined in the text. Words and phrases that are emphasized also appear in *italics*.

Monospace

Code examples, output, and system messages appear in a monospace font.

ALLCAPS

Tivoli NetView for OS/390 commands appear in ALL CAPS.

Contacting Customer support

For support for this or any other Tivoli product, you can contact Tivoli Customer Support in one of the following ways:

- Visit our Web site at **www.tivoli.com/support**
- Send e-mail to **support@tivoli.com**

Customers in the United States can also call 1-800-TIVOLI8 (1-800-848-6548). International customers should consult the Web site for customer support telephone numbers. You can also review the Customer Support Handbook, which is available at:
www.tivoli.com/support/handbook/

When you contact Tivoli Customer Support, be prepared to provide identification information for your company, so that support personnel can assist you more readily. Company identification information may also be needed to access various online services available on the Web site.

The Web site offers extensive information, including a guide to support services (the Customer Support Handbook); frequently asked questions (FAQs); and documentation for all Tivoli products, including Release Notes, Redbooks, and Whitepapers. The documentation sets for some product releases are available in both PDF and HTML formats. Translated documents are also available for some product releases.

You can order documentation by e-mail at **swdist@tivoli.com**. Please provide the part number or order number of the desired document; alternatively, you can provide the document's title, version number, and date of publication.

We are very interested in hearing about your experience with Tivoli products and documentation. We also welcome your suggestions for improvements. If you have comments or suggestions about our documentation, please contact us in one of the following ways:

- Send e-mail to: **pubs@tivoli.com**
- Fill out our customer feedback survey at: **www.tivoli.com/support/feedback**

1

Introducing TBSM

Tivoli Business Systems Manager (TBSM) is a business systems management tool that enables you to perform distributed management, OS/390 management, or both. Even when a business system spans multiple platforms, TBSM enables you to graphically monitor and control interconnected business components and operating system resources. A business component and its resources are referred to as a Line of Business (LOB).

Using the LOB concept, TBSM helps you plan, define, and control your business system. TBSM, together with other Tivoli management business components, helps you manage the dependencies between business components and their underlying infrastructure.

A Line of Business (LOB) consists of:

- All business components that together perform a specific business function
- The defined relationships between business components
- The measures that determine whether the LOB is functioning properly

In today's environment, an LOB may span both the mainframe and distributed platforms and may contain a collection of applications or business components that run on a variety of platforms. For example a banking system LOB designed to support transactions over the Web, typically includes the following:

- A Web server running outside the company's intranet and connected directly to the Internet
- A firewall that provides secure connectivity to a machine running a custom business component, such as loan processing

The loan processing business component usually runs on a distributed platform and interfaces to business components running on a mainframe computer. The mainframe handles all the bank transactions. This LOB presents challenges to a system manager because it crosses the typically isolated environments of host and distributed systems.

Another example of a LOB is an e-mail system. An e-mail LOB includes all the instances of e-mail business components that are being used in your network. You could have a mix of Lotus Notes servers and clients, POP mail or Microsoft Exchange servers and clients, and other e-mail business components. An e-mail LOB includes definitions that tell whether each entity in the LOB is a server, a client, or both. It also includes definitions of the monitors that collect status information for each business component in the LOB, as well as definitions of the relationships between business components in a LOB.

TBSM can help you manage:

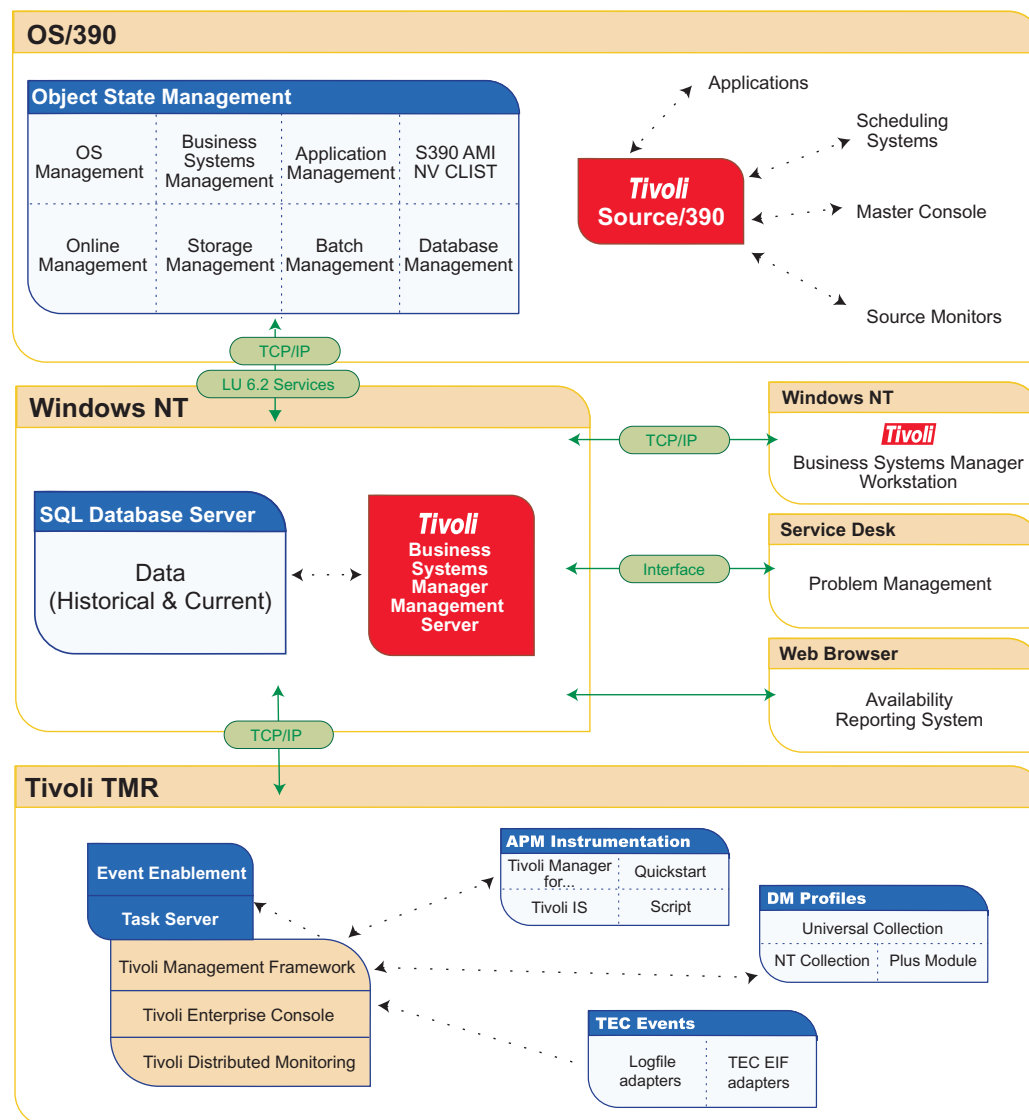
-
- Business components that communicate across multiple systems
 - Distributed business components, OS/390-based business components, or both
 - IMS installations
 - CICSplex SM installations
 - DB2 installations
 - Middleware, MQSeries, and databases such as Oracle, using modules such as Tivoli Manager for MQSeries, and Tivoli Manager for Oracle
 - Custom applications.

TBSM Architecture

TBSM architecture resides in a three-tiered client/server environment that has server components in both the OS/390 and Windows NT Server environments. Because server components exist within both the OS/390 and Windows environments, TBSM uses both SNA and TCP/IP as transport mechanisms for data.

The TBSM client workstation supports Windows NT and Windows 2000. The following diagram illustrates the architecture for TBSM Enterprise Edition.

Tivoli Business Systems Manager - Enterprise Edition Architecture



Object State Management

TBSM monitors for state changes that may occur in the various resources within your corporation's environment. This includes any changes to operating systems, applications, online transactions, batch processes, and database and storage management.

Tivoli Source/390

Tivoli Source/390 is a set of components that runs under IBM's OS/390 host operating environment. Source/390 collects data from the various sources and packages it so that it can be sent to the TBSM Management Server.

The Tivoli Source/390 set of components is part of the Enterprise Edition only.

SQL Database Server

The TBSM SQL Database Server stores all availability information. This TBSM database contains both an historical database and a current database.

TBSM Management Server

The TBSM Management Server processes all the availability data that is collected from various sources. Availability information is inserted in the TBSM database, where intelligent agents provide alerts on monitored objects and then broadcast those alerts to TBSM workstations. The TBSM Management Server handles all user requests from the workstation and includes a database server that is built around a Microsoft SQL Server database.

TBSM Workstation

The TBSM workstation displays objects in customized views called Line of Business Views. Objects are presented in a hierarchical TreeView so that users can visualize the relationship between objects. Alerts are overlaid on the objects when their availability is threatened.

Service Desk

Integration with problem management systems, such as the Tivoli Service Desk products, is provided. For additional information, see Tivoli Business Systems Manager TSD Integration Configuration/Installation Guide.

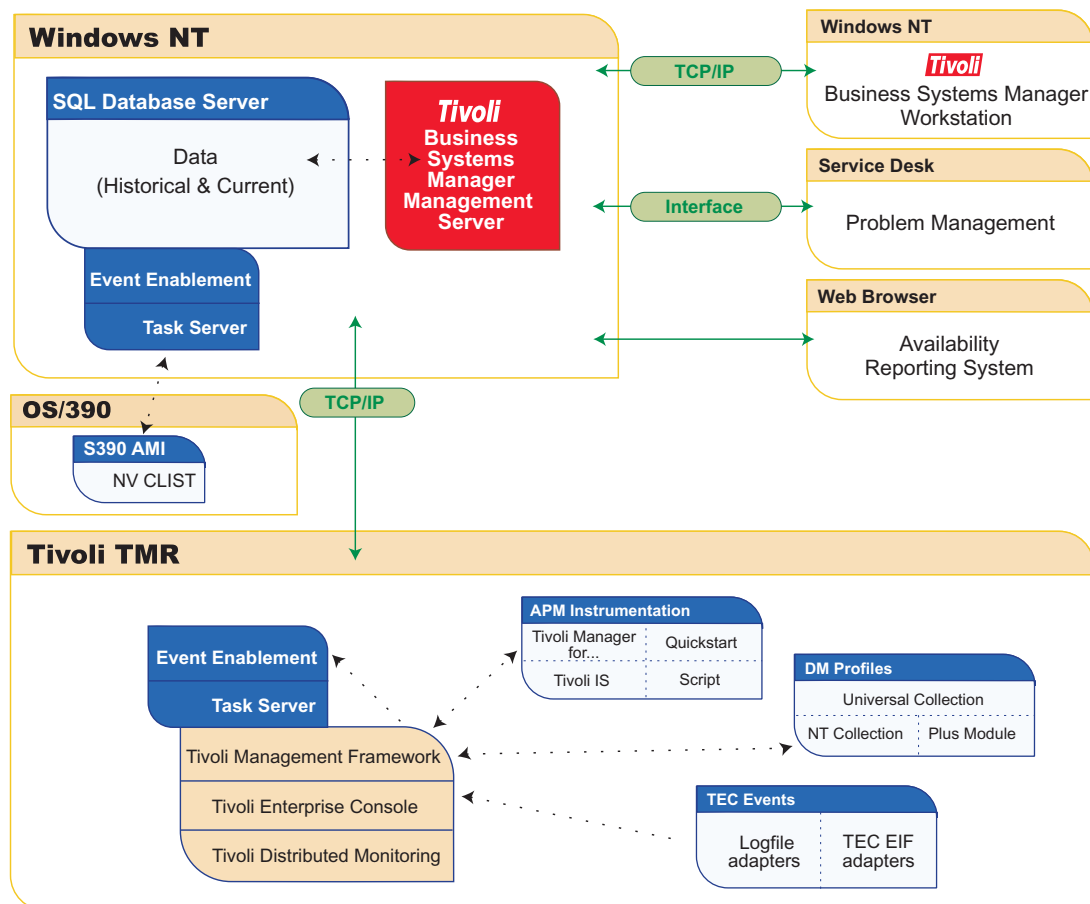
Availability Reporting System

Availability information can be viewed using a standard Web browser. Reports can be generated so that you can analyze trends and perform capacity planning.

TBSM - Distributed Architecture

The following diagram illustrates the TBSM - Distributed Edition (TBSM/D) architecture. TBSM/D does not include the Source/390 components.

Tivoli Business Systems Manager - Distributed Edition Architecture



Event Enablement

Tivoli Event Enablement is installed on the Tivoli Enterprise Console (TEC) server. It enables the TEC to forward events to TBSM. Tivoli Event Enablement defines event classes and rules for handling TBSM/D-related events. The rules provided for TBSM/D event classes enable the TEC to communicate the events to one or more TBSM agent listeners. Events sent from the TEC to TBSM enable TBSM/D to accurately reflect the current status of business components and any associated monitors and connections. When the TBSM Agent Listener is down, Tivoli Event Enablement caches events to avoid loss of information. Once communication is restored, the TEC server and the TBSM/D database are quickly re-synchronized.

Task Server

The Task Server interacts with the Tivoli Framework (TMF) and the Tivoli NetView for OS/390 environments. Command requests from an operator are routed to the task server, which then routes the requests to the appropriate environment for execution. The responses are returned to the task server where they are correlated to the request and then routed back to the requesting operator. The task server communicates with the Agent Listener using

TCP/IP, it communicates with Tivoli NetView for OS/390 using either TCP/IP or LU 6.2, and it communicates with TMF using native framework APIs.

TMF tasks are issued to the framework using the operator id that the task server is running under. Under NT, the operator id that the task server is running under can be set via the Services applet in the Control Panel. Under UNIX, the task server runs as root. The ID that the task server is running under must be set by a Tivoli Administrator. Tivoli NetView for OS/390 commands are issued using the Tivoli NetView operator id and password that is provided by definitions in the SQL server.

Instrumentation

Instrumentation provides information about specific applications, system software, or business components to a management system, such as TBSM/D. You can instrument business components in either a Line of Business or a standalone application.

2

Installing TBSM Base Services

This chapter describes the installation tasks for the Tivoli Business Systems Manager's (TBSM's) Base Services, which are the services and components that are used for both the Distributed Edition and the Enterprise Edition of TBSM. These tasks must be performed before using either the Distributed Edition or the Enterprise Edition.

For the Enterprise Edition, install the Base services and components of TBSM on different Windows servers, as indicated in the hardware and software requirements listed in Appendix B.

For the Distributed Edition, install the Base services and components of TBSM on two servers, as indicated in the hardware and software requirements listed in Appendix C.

Exit all Windows programs before beginning the **Setup** program.

Upon clicking the **Setup.exe** icon, TBSM prepares the Install Shield Wizard that guides you through the installation process. Once the Install Shield Wizard is ready, the **Welcome to the Tivoli Business Systems Manager Setup Program** dialog opens.

To **Exit** the **Setup** program at any time before installation is complete, click **Cancel**. To continue with each step of the install, click **Next**.

Language Support

In addition to English, the following languages are supported in this release of TBSM:

- Brazilian Portuguese
- German
- French
- Italian
- Japanese
- Korean
- Simplified Chinese
- Spanish
- Traditional Chinese

After TBSM installation is complete, refer to the README file on the language pack CD for information about updating TBSM for a supported language other than English.

Installing SQL Database Server

To install the SQL Database Server: (If the SQL Database Server is already installed, uninstall it and install it using these instructions to set the correct order for TBSM.)

1. Insert the **Microsoft SQL Server 7.0** CD and select **Setup.bat**.
2. The **Microsoft SQL Server Version 7.0** setup dialog begins. The Install Shield Wizard guides you through the installation. Follow the instructions in the dialogs.
3. In the **Setup Type** dialog, select **Custom** and enter the destination folders for the **Program Files** and **Data Files**. If the default is **D:\MSSQL7** for Destination Folders and for Program Files, leave it as is. Otherwise, click the Browse button and use **D:\MSSQL7** as the destination folder for the **Program Files** and **Data Files**. Click **Next**.

The default drive on which to install the SQL database server is **D:**. If you use a different drive, you must change the SQL script used to attach the TBSM databases to reflect the correct drive. See “Installing Tivoli Business Systems Manager Enterprise Edition” or “Installing Tivoli Business Systems Manager Distributed Edition” for additional information.

4. When the **Select Components** dialog opens, click **Next** to accept the defaults (components with a check mark in the box).
5. On the **Character Set/Sort Order/Unicode Collation** dialog, use **1252/ISO** for **Character Set** and select **Binary Order** as the **Sort Order**. Keep the remaining defaults. Click **Next**.
6. On the **Network Libraries** dialog, keep the defaults for **Named Pipe name** and **TCP/IP Sockets**. Select **Multi-Protocol** and **Enable Multi-Protocol encryption**. Click **Next**.
7. On the **Services Accounts** dialog, keep the default: **Use the same account for each service**. For **Service Settings**, select **Use the Local System account**. Click **Next**.
8. When the **Start Copying Files** dialog opens, click **Next** to continue with the SQL Server Install. If you want to change any parameters, click **Back**.
9. On the **Choose Licensing Mode** dialog, select **Per Server**. Click **Add Licenses**. Click **Continue**.
10. On the **New Client Access License** dialog, type **200** in the **Quantity** field and click **OK**.
11. When the **Per Server Licensing** dialog opens, read the text and if you agree to the conditions, select the box next to **I agree that:** and click **OK**. (Selecting the box puts a check mark in it.)
12. When the **Choose Licensing Mode** dialog reopens, choose your licensing mode (we recommend **Per Server**) and click **Continue** to complete the installation.
13. Continue the installation and click **Finish** on the **Setup Complete** dialog.

Installing SQL Server 7.0 Service Pack 2

SQL Server 7.0 Service Pack 2 is available from Microsoft’s Internet Web site.

To install SQL Server 7.0 Service Pack 2.0:

1. From the **X86/Setup** directory, run **setupsql.exe**. The **Microsoft SQL Server Service Pack 2** setup dialog begins. The Install Shield Wizard guides you through the installation. Accept the defaults throughout the installation.
2. When the **Setup Complete** dialog appears. Select **Yes, I want to restart my computer now**. Click **Finish** to conclude this installation.

Changing the SQL Server Authentication Password

After the SQL Server 7.0 and SQL Server 7.0 Service Pack 2 are installed, you must change the **SQL Server authentication password** for the 'sa' account. Do this from the **SQL Server Enterprise Manager** by selecting **Start --> Programs --> Microsoft SQL Server 7.0 --> Enterprise Manager**.

1. In the **Enterprise Manager** window, expand the **SQL Server Group** list.
2. Select your SQL server (usually the only SQL Server listed).
3. Expand the **Security** list.
4. From the list of security options, select **Logins**.
5. In the right pane of the window that opens, all available user accounts are listed. Right-click on the 'sa' account and select **Properties**.
6. Type the new 'sa' password and click **Apply**.
7. When prompted to confirm the new password, type the 'sa' password again and click **OK** and the dialog closes.

Turning Off Certain Client-Side Options

Certain SQL client-side options must be manually turned off in SQL 7.0 for TBSM to function correctly.

Turn off the option **Use ANSI NULLs, padding and warnings** (in two places on the menus) in SQL Server 7.0's Query Analyzer Tool (ISQLW). To do this:

■ Step 1

1. Start SQL 7.0's **Query Analyzer** by selecting **Tools --> SQL Server Query Analyzer**.
2. From the Query analyzer window, select **File --> Configure**.
3. Select the **New Connections** tab on the property sheet.
4. Uncheck **Use ANSI NULLs, padding and warnings**.
5. Click **OK** to close the dialog.

■ Step 2

1. Select **Query --> Current Connection Options**.
2. On the **General** tab, uncheck **Use ANSI NULLs, padding and warnings**.
3. Click **OK** to close the dialog.

Close **Enterprise Manager** and **SQL Server Query Analyzer**.

Installing TBSM Base Components

If you're installing the Enterprise Edition, follow the steps in the section titled "Installing TBSM Enterprise Edition." If you're installing the Distributed Edition, follow the steps in the section titled "Installing TBSM Distributed Edition."

Installing TBSM Enterprise Edition

Following are instructions for installing the TBSM Enterprise Edition components.

Installing the Database Server

The first TBSM component to install is the database server. To install the database server:

1. Insert the **TBSM Base Services** CD in the Database Server computer.
 2. Double-click the **Setup.exe** icon.
 3. When the System File Upgrade dialog opens, click **Next** on the **Welcome** dialog.
 4. On the **Choose Destination Location** dialog, we recommend that you place these files in a temporary directory. Choose the **Browse** button to select a directory (or use the default directory). Once a desired directory is selected or to use the default, click **Next**.
 5. When the **System File Upgrade** is complete, click **Finish**. If you are prompted to reboot the computer, do so now.
 6. The **Welcome to the Tivoli Business Systems Manager Setup** program dialog opens. The text in this dialog is basic information about the Setup program, including how to discontinue the install. Read the information and click **Next**.
 7. When the **Choose Destination Location** dialog opens, select a directory to install the Tivoli Business Systems Manager application and click the **Browse** button. The selected path is displayed in the **Destination Folder**. Once the desired path is selected or to choose the default destination folder, click **Next**.
 8. On the **Setup Type** dialog, choose **Custom** and click **Next**.
 9. When the **Select Components** dialog opens, select the following components for an **SQL Extensions** install:
 - **Workstation Program Files**
 - **Help Files**
 - **SQL Extension Files**
 - **Tools and Utilities**
 - **Staged Event Loader**
- Note:** If you also want the Distributed components, select the following options in addition to those listed above:
- **NT Agent Listener**
 - **TBSM xdfparser**
 - **Event Enablement** (This is automatically selected.)
10. Click **Next**.
 11. Enter the **hostname** of the local server machine. (By default, the hostname should be in the hostname field. If it is not, enter it.) Click **Next**.

12. Enter the **hostname** of the **SQL Server** machine, the SQL Administrator **Username**, and SQL Administrator **Password**. Click **Next**.
13. Enter the host name of the application server machine.
14. On the **Select Program Folder** dialog, select the **Program Folder** to host all application files, or accept the default. Click **Next**.
15. When the **Setup Complete** dialog opens, if you are prompted to restart, choose **No**. Click **Finish**.
16. After installing the Database server, the following services should be on this computer. Determine if these services are on the computer by choosing **Start-->Programs-->Administrative Tools-->Server Manager-->Computer-->Services**.
 - **Tivoli BSM Batch Rule Server**
 - **Tivoli BSM Database Validator**
 - **Tivoli BSM Notification Server**
 - **Tivoli BSM Propagation Agent Dispatcher**
 - **Tivoli BSM Event Enablement**
 - **Tivoli BSM Staged Event Loader**
 - **Tivoli BSM TSD Event Handler**

Note: If you also selected the Distributed components, the following additional servers will be available:

 - **Agent Listener**
 - **Task Server**
17. While in **Server Manager**, stop the **SQL Server Agent** and the **MSDTC Server** service.
18. Ensure that the **MSSQL7** Server service is started.
19. Apply `<install_dir>\SQL\DetachDatabases.sql` to your database server. (*.LDF files will remain on your drive until AttachDatabases.sql is run later in this procedure.)
20. Expand the database devices from TBSM Base Services CD (DBDevices\DBDevices_V1.5.exe) to the **D:\MSSQL7\Data** sub-directory. (These are database devices, not backup devices and they can be attached to the server without you having to create database devices, create the databases, and perform database restores.) You may extract the files by changing to the target directory and running the executable.
 - **D:\MSSQL7\Data>E:\DBDevices\DBDevices_V1.5.exe**
21. Edit a copy of `<install_dir>\SQL\AttachDatabases.sql` and make the following changes as necessary.
 - **SQLServer 7.0 installation directory (D:\MSSQL7\ is the default).** If the SQL server is installed on a drive other than **D:**, modify the script as necessary to point to the appropriate drive where the SQL server is installed.
 - **Hostname of the database machine (Replace XXXX in the SQL code.)**
 - **Hostname of the propagation agent machine (Replace YYYY in the SQL code.)**

■ **Hostname of the web server machine (Replace ZZZZ in the SQL code.)**

22. Apply the edited copy of <install_dir>\SQL\AttachDatabases.sql to the database server. (The .LDF files that may or may not exist are deleted by this procedure before the new databases are attached.)
23. Apply <install_dir>\SQL\SchemaTypes.sql to the master database.
24. Apply <install_dir>\SQL\SchemaTypes.sql to the model database.
25. Apply <install_dir>\SQL\SchemaTypes.sql to the tempdb database.
26. Apply <install_dir>\SQL\MasterDBChanges.sql to your master database server. If the SQL server is installed on a drive other than D:\, modify the script as necessary to point to the appropriate drive where the SQL server is installed.
27. Apply <install_dir>\SQL\SchemaMessages.sql to your master database server.
28. Restart the computer.

Installing the History Server

To install the History Server:

1. Insert the **TBSM Base Services** CD in the History Server computer.
2. Double-click the **Setup.exe** icon.
3. When the **System File Upgrade** dialog opens, read the text and click **Next**.
4. The **Choose Destination Location** dialog opens. It is recommended that you place these files in a Temporary directory. Choose the **Browse** button to select a directory. Once a desired directory is selected, click **Next**.
5. When the **System File Upgrade** is complete, click **Finish**. If you are prompted to reboot the computer, do so now.
6. On the **Setup Type** dialog, choose **Custom** and click **Next**.
7. Follow instructions on the installation dialogs to continue the installation.
8. On the **Select Components** dialog, select the following components:
 - **Workstation Program Files**
 - **Help Files**
 - **Reporting System**
 - **Active Documentation**
 - **Tools and Utilities**
9. Click **Next**.
10. Enter the **hostname** of the local server machine. (By default, the hostname should be in the hostname field, If it is not, enter it.) Click **Next**.
11. Enter the **hostname** of the **TBSM Database Server** machine, the SQL Administrator **Username**, and SQL Administrator **sa Password**. Click **Next**.
12. Enter the **name** of the **Application Server** machine. Click **Next**.
13. The **Select Program Folder** dialog box opens. Select the *Program Folder* to host all applications. Click **Next**.

-
14. On the **Setup Complete** dialog, when asked if you want to restart the computer choose **No** and click **Finish**.

Configuring the Reporting System

The TBSM Reporting System is one of the components you selected when you installed the History server, and it must be configured to point to the appropriate SQL Server databases.

To configure the TBSM Reporting System:

1. Go to **Start-->Programs-->Tivoli-->Reporting System Database Configuration**.
2. Expand the drop-down menu for **Select Connection** and select **WebServer**. The **WebServer** connection must point to the **WebServer** database.
3. Enter the following information on the TBSM Reporting Database Configuration dialog:
 - **Select Connection = WebServer**
 - **Connection Name = WebServer**
 - **Description = Reporting Menu / User Database**
 - **Driver = {SQL Server}**
 - **Server Name = database_server_name**
 - **User ID = sa**
 - **Password = (the password)**
 - **Database Name = WebServer**
 - **Connection Timeout = 600**
4. Click **Test** to test the connection. Clicking **Update** automatically tests the connection before updating it.

Configure the DefaultData in a similar manner:

1. Expand the drop-down menu for **Select Connection** and select **DefaultData**. The **DefaultData** connection must point to the **Object** database from which the history data will be obtained.
2. Enter the following information on the TBSM Reporting Database Configuration dialog:
 - **Select Connection = DefaultData**
 - **Connection Name = DefaultData**
 - **Description = Default History Database**
 - **Driver = {SQL Server}**
 - **Server Name = database_server_name**
 - **User ID = sa**
 - **Password = (the password)**
 - **Database Name = Object**
 - **Connection Timeout = 1200**

Configure the **TBSM Documentation** in a similar manner:

-
1. Expand the drop-down menu for **Select Connection** and select **TBSM Documentation**. The **TBSM Documentation** connection must point to the **Object** database from which the history data will be obtained.
 2. Enter the following information on the TBSM Reporting Database Configuration dialog:
 - **Select Connection = TBSMDocumentation**
 - **Connection Name = TBSMDocumentation**
 - **Description = TBSM Server for Documentation**
 - **Driver = {SQL Server}**
 - **Server Name = database_server_name**
 - **User ID = sa**
 - **Password = (the password)**
 - **Database Name = Object**
 - **Connection Timeout = 600**

Installing the Application Server

Before installing the Application server make the following changes to the machine on which it is to be installed:

1. Make changes to the system security:
 - a. Add two new Groups in the User Management section of the system security:
 - TBSM_Administrators
 - TBSM_Operators

These groups reflect the security options for the TBSM JAVA client. TBSM users requiring administrative access must be added to the group TBSM_Administrators. All other TBSM users should be added to the TBSM_Operators group. If a user attempts to access the TBSM JAVA console, and is not in one of these authorized groups, they will receive an error, **User name xxx could not be validated**.

- b. The Java Console Application server service must be running under an account that has two system privileges:
 - act as part of the operating system
 - replace a process level token

If you are granting these rights to the login under which you are currently logged on, then you must log off and back on in order for these newly-granted rights to take effect.

To install the Application Server:

1. Insert the **TBSM Base Services** CD in the Application Server computer.
2. Double-click the **Setup.exe** icon.
3. The System File Upgrade dialog opens. Read the text and click **Next**.

4. For the **Choose Destination Location** dialog, it is recommended that you place these files in a Temporary directory. Choose the **Browse** button to select your directory. Once a desired directory is selected, click **Next**.
5. When the **System File Upgrade** is complete, click **Finish**. If you are prompted to reboot the computer, do so now.
6. On the **Setup Type** dialog, choose **Custom** and click **Next**.
7. Follow instructions on the installation dialogs to continue the installation.
8. On the **Select Components** dialog, select the following components:
 - **Workstation Program Files**
 - **Help Files**
 - **Application Server Files**
 - **Java Application Server**
 - **Tools and Utilities**

Note: If you are currently running TBSM 1.1.1, you also need to install **TBSM 1.1.1 Compatibility Services**. Do this by selecting **TBSM 1.1.1 Compatibility Services** on the **Select Components** dialog. This component enables TBSM 1.1.1 Workstations to communicate with the TBSM 1.5 server(s).

Select these 2 components:

- Tivoli BSM 1.1.1 Application Server
- Tivoli BSM 1.1.1 Notification Server

If the TBSM workstations and servers communicate through a firewall, the ports chosen for TBSM 1.5 should be different from the ports used previously in TBSM 1.1.1. **TBSM 1.1.1 Compatibility Services** should then be configured to use the TBSM 1.1.1 port settings and the new TBSM 1.5 services configured to use new port numbers.

After you migrate to TBSM 1.5, disable the TBSM 1.1.1 Compatibility Services (Tivoli BSM 1.1.1 Application Server and Tivoli BSM 1.1.1 Notification Server) from the control panel on the Application Server.

9. Click **Next**.
10. Enter the **hostname** of the local server machine. (By default, the hostname should be in the hostname field. If it is not, enter it.) Click **Next**.
11. On the parameters for the SQL Server machine, enter the **hostname** of the **SQL Server** machine, the SQL Administrator **Username**, and SQL Administrator **Password**. Click **Next**.
12. Enter the Web alias to be used by the Java Application server. The recommended alias is **TBSM**. Click **Next**.
13. Enter the **name** of the **Application Server** machine. Click **Next**.
14. On the **Select Program Folder** dialog, select the **Program Folder** to host all application files. Click **Next**.

-
15. On the **Setup Complete** dialog, when asked if you want to restart the computer, choose **No** and click **Finish**.

After installing the Application server, the following services should be on this computer:

- **Tivoli BSM Console Server**
- **Tivoli BSM Application Server**
- **Tivoli BSM Database Validator**

Installing the Propagation Agent Server

To install the Propagation Agent server computer:

1. Insert the **TBSM Base Services** CD in the Propagation Server computer.
2. Double click the **Setup.exe** icon.
3. When the System File Upgrade dialog opens, read the text and click **Next**.
4. On the **Choose Destination Location** dialog, it is recommended that you place these files in a Temporary directory. Choose the **Browse** button to select your directory. Once a desired directory is selected, click **Next**.
5. When the **System File Upgrade** is complete, click **Finish**. If you are prompted to reboot the computer, do so now.
6. On the **Setup Type** dialog, choose **Custom** and click **Next**.
7. Follow instructions on the installation dialogs to continue the installation.
8. On the **Select Components** dialog, select the following components:
 - **Workstation Program Files**
 - **Help Files**
 - **Propagation Agent Components**
 - **Tools and Utilities**
9. Click **Next**.
10. Enter the **hostname** of the local server machine. (By default, the hostname should be in the hostname field, If it is not, enter it.) Click **Next**.
11. Enter the **hostname** of the **SQL Server** machine, the SQL Administrator **Username**, and SQL Administrator **Password**. Click **Next**.
12. Enter the **name** of the **Application Server** machine. Click **Next**.
13. On the **Select Program Folder** dialog, select the **Program Folder** to host all application files. Click **Next**.
14. On the **Setup Complete** dialog when asked if you want to restart the computer, choose **No** and click **Finish**.

After installing the Propagation Agent server, the following services should be on this computer:

- **Tivoli BSM Enqueue Proxy Server**
- **Tivoli BSM Database Validator**
- **Tivoli BSM Remote Execution Server**

Installing Event Handler Server

To install the Event Handlers:

1. Insert the **TBSM Base Services** CD in the Event Handler computer.
2. Double-click the **Setup.exe** icon.
3. When the System File Upgrade dialog opens, read the text and click **Next**.
4. On the **Choose Destination Location** dialog, it is recommended that you place these files in a Temporary directory. Choose the **Browse** button to select your directory. Once a desired directory is selected, click **Next**.
5. When the **System File Upgrade** is complete, click **Finish**. If you are prompted to reboot the computer, do so now.
6. On the **Setup Type** dialog, choose **Custom** and click **Next**.
7. Follow instructions on the installation dialogs to continue the installation.
8. On the **Select Components** dialog, select the following components:
 - **Workstation Program Files**
 - **Help Files**
 - **Mainframe Monitoring Components**
 - **Tools and Utilities**
9. Click **Next**.
10. Enter the **hostname** of the local server machine. (By default, the hostname should be in the hostname field, If it is not, enter it.) Click **Next**.
11. Enter the **hostname** of the **SQL Server** machine, the SQL Administrator **Username**, and SQL Administrator **Password**. Click **Next**.
12. Enter the **name** of the **Application Server** machine. Click **Next**.
13. On the **Select Program Folder** dialog, select the **Program Folder** to host all application files. Click **Next**.
14. On the **Setup Complete** dialog when asked if you want to restart the computer, choose **No** and click **Finish**.

After installing the Event Handlers server, the following services should be on this computer:

- **Tivoli BSM Database Validator**
- **Tivoli BSM Enqueue Proxy Server**
- **Tivoli BSM MVSIPSLListener**
- **Tivoli BSM MVSUpload Rule Server**

Installing the Traditional TBSM Client

To install the traditional TBSM Client workstation:

1. Insert the **TBSM Base Services** CD in the Client Workstation.
2. Double-click the **Setup.exe** icon.
3. When the **System File Upgrade** dialog opens, read the text and click **Next**.

-
4. On the **Choose Destination Location** dialog, it is recommended that you place these files in a Temporary directory. Choose the **Browse** button to select the directory. Once a desired directory is selected, click **Next**.
 5. When the **System File Upgrade** is complete, click **Finish**. If you are prompted to reboot the computer, do so now.
 6. On the **Setup Type** dialog, choose **Custom** and click **Next**.
 7. Follow instructions on the installation dialogs to continue the installation.
 8. On the **Select Components** dialog, select the following components:
 - **Workstation Program Files**
 - **Help Files**
 - **Tools and Utilities**
 9. Click **Next**.
 10. Enter the **hostname** of the local server machine. (By default, the hostname should be in the hostname field, If it is not, enter it.) Click **Next**.
 11. Enter the **name** of the **Application Server** machine. Click **Next**.
 12. On the **Select Program Folder** dialog, select the **Program Folder** to host all application files. Click **Next**.
 13. On the **Setup Complete** dialog when asked if you want to restart the computer, choose **Yes** and click **Finish**.

Installing the Java Console Client

The install process uses the /TMP directory as a working directory. If there isn't enough room in that directory, you can install using the following command:

```
setup -is:tempdir <name of another directory>
```

To install the Java Console :

1. Run the Java Client Setup program located in the ConsoleInstalls\Win32 directory on the TBSM installation CD to access the Java installation wizard.
2. When the installer dialog opens, select the directory for installation. The installation directories are:
 - 5-BSM-0002.tar - TBSM Console (aix_ia64) - for Intel 64 platform
 - 5-BSM-0003.tar - TBSM Console (aix_power) - for Power PC platform
 - 5-BSM-0004.tar - TBSM Console (linux) - for Linux platform
 - 5-BSM-0005.tar - TBSM Console (sol_sparc) - for Solaris Sparc platform
 - 5-BSM-0006.tar - TBSM Console (sol_x86) - for AMD 86 platform
 - 5-BSM-0007.tar - TBSM Console (win32) - for Windows platforms
3. Enter the fully qualified machine name of the TBSM Application server.
4. After the installation process completes, press **Finish**.

Note: For Unix platforms, ensure that you have applied the current patches for the operating system that are required to run Java 1.3. To get those patches, go to the Web site of the vendor for that platform and download the patches and install them before installing the Java console.

Restarting the Servers

Reboot the SQL Database Server first. After the SQL Database Server is up and running, reboot the remaining servers in any order. **However, if you are using a SNA server, reboot it first after you reboot the SQL Database Server.** (For information about Installing and Configuring the Microsoft SNA Server, see Chapter 7 of this document.)

Installing TBSM Distributed Edition

This section explains how to install the Distributed Edition of TBSM.

Installing the Database Server Components

Install the TBSM components on the database server machine:

1. Insert the **TBSM Base Services** CD in the Database Server computer.
2. Double-click the **Setup.exe** icon.
3. When the System File Upgrade dialog opens, click **Next** on the **Welcome** dialog.
4. On the **Choose Destination Location** dialog, we recommend that you place these files in a temporary directory. Choose the **Browse** button to select a directory (or use the default directory). Once a desired directory is selected or to use the default, click **Next**.
5. When the **System File Upgrade** is complete, click **Finish**. If you are prompted to reboot the computer, do so now.
6. The **Welcome to the Tivoli Business Systems Manager Setup** program dialog opens. The text in this dialog is basic information about the Setup program, including how to discontinue the install. Read the information and click **Next**.
7. When the **Choose Destination Location** dialog opens, select a directory to install the Tivoli Business Systems Manager application and click the **Browse** button. The selected path is displayed in the **Destination Folder**. Once the desired path is selected or to choose the default destination folder, click **Next**.
8. On the **Setup Type** dialog, choose **Custom** and click **Next**.
9. When the **Select Components** dialog opens, select the following components for database server for distributed edition:
 - **SQL Extension Files**
 - **Tools and Utilities**
 - **Staged Event Loader**
 - **NT Agent Listener**
 - **TBSM xdfparser**
 - **Event Enablement** (This is automatically selected.)

Optionally, you may include:

- **Workstation Program Files** and **Help Files** components if you wish to use the TBSM client on the database server computer

-
- **Reporting System** and **Active Documentation** components if you wish to have a history server
10. Click **Next**.
 11. Enter the **hostname** of the local server machine. (By default, the hostname should be in the hostname field. If it is not, enter it.) Click **Next**.
 12. Enter the **hostname** of the **SQL Server** machine, the SQL Administrator **Username**, and SQL Administrator **Password**. Click **Next**.
 13. Enter the name of the **Application Server** machine.
 14. On the **Select Program Folder** dialog, select the **Program Folder** to host all application files, or accept the default. Click **Next**.
 15. When the **Setup Complete** dialog opens, if you are prompted to restart, choose **No**. Click **Finish**.
 16. After installing the Database server, the following services should be on this computer. Determine if these services are on the computer by choosing **Start-->Programs-->Administrative Tools-->Server Manager**.
 - **Tivoli BSM Agent Listener**
 - **Tivoli BSM Batch Rule Server**
 - **Tivoli BSM Database Validator**
 - **Tivoli BSM Event Enablement**
 - **Tivoli BSM Notification Server**
 - **Tivoli BSM Propagation Agent Dispatcher**
 - **Tivoli BSM Staged Event Loader**
 - **Tivoli BSM Task Server**
 - **Tivoli BSM TSD Event Handler**
 17. While in **Server Manager**, stop the **SQL Server Agent** and the **MSDTC Server Service**.
 18. Ensure that the **MSSQL7 Server** service is started.
 19. Apply `<install_dir>\SQL\DetachDatabases.sql` to your database server. (*.LDF files will remain on your drive until AttachDatabases.sql is run later in this procedure.)
 20. Expand the database devices from TBSM Base Services CD (`DBDevices\DBDevices_V1.5.exe`) to the **D:\MSSQL7\Data** sub-directory. (These are database devices, not backup devices and they can be attached to the server without you having to create database devices, create the databases, and perform database restores.) You may extract the files by changing to the target directory and running the executable.
 - **D:\MSSQL7\Data>E:\DBDevices\DBDevices_V1.5.exe**
 21. Edit a copy of `<install_dir>\SQL\AttachDatabases.sql` and make the following changes as necessary.
 - **SQLServer 7.0 installation directory (D:\MSSQL7\ is the default).** If the SQL server is installed on a drive other than **D:**, modify the script as necessary to point to the appropriate drive where the SQL server is installed.
-

- **Hostname of the database machine (Replace XXXX in the SQL code)**
 - **Hostname of the propagation agent machine (Replace YYYY in the SQL code)**
 - **Hostname of the web server machine (Replace ZZZZ in the SQL code)**
22. Apply the edited copy of <install_dir>\SQL\AttachDatabases.sql to your database server. (The .LDF files that may or may not exist are deleted by this procedure before the new databases are attached.)
 23. Apply <install_dir>\SQL\SchemaTypes.sql to the master database.
 24. Apply <install_dir>\SQL\SchemaTypes.sql to the model database.
 25. Apply <install_dir>\SQL\SchemaTypes.sql to the tempdb database.
 26. Apply <install_dir>\SQL\MasterDBChanges.sql to your master database server. If the SQL server is installed on a drive other than D:\, modify the script as necessary to point to the appropriate drive where the SQL server is installed.
 27. Apply <install_dir>\SQL\SchemaMessages.sql to your master database server.
 28. Restart the computer.

Configuring the Reporting System

If you installed the TBSM Reporting System, you must configure it to point to the appropriate SQL Server databases.

To configure the TBSM Reporting System:

1. Go to **Start-->Programs-->Tivoli-->Reporting System Database Configuration**.
2. Expand the drop-down menu for **Select Connection** and select **WebServer**. The **WebServer** connection must point to the **WebServer** database.
3. Enter the following information on the TBSM Reporting Database Configuration dialog:
 - **Select Connection = WebServer**
 - **Connection Name = WebServer**
 - **Description = Reporting Menu / User Database**
 - **Driver = {SQL Server}**
 - **Server Name = database_server_name**
 - **User ID = sa**
 - **Password = (the password)**
 - **Database Name = WebServer**
 - **Connection Timeout = 600**
4. Click **Test** to test the connection. Clicking **Update** automatically tests the connection before updating it.

Configure the **TBSM Documentation** in a similar manner:

1. Expand the drop-down menu for **Select Connection** and select **TBSM Documentation**. The **TBSM Documentation** connection must point to the **Object** database from which the history data will be obtained.
2. Enter the following information on the TBSM Reporting Database Configuration dialog:

-
- **Select Connection = TBSMDocumentation**
 - **Connection Name = TBSMDocumentation**
 - **Description = TBSM Server for Documentation**
 - **Driver = {SQL Server}**
 - **Server Name = database_server_name**
 - **User ID = sa**
 - **Password = (the password)**
 - **Database Name = Object**
 - **Connection Timeout = 600**

Installing the Application Server Components

Before installing the Application server make the following changes to the machine on which it is to be installed:

1. Make changes to the system security:
 - a. Add two new Groups in the User Management section of the system security:
 - TBSM_Administrators
 - TBSM_Operators

These groups reflect the security options for the TBSM JAVA client. TBSM users requiring administrative access must be added to the group TBSM_Administrators. All other TBSM users should be added to the TBSM_Operators group. If a user attempts to access the TBSM JAVA console, and is not in one of these authorized groups, they will receive an error, **User name xxx could not be validated.**

- b. The Java Console Application server service must be running under an account that has two system privileges:
 - act as part of the operating system
 - replace a process level token

If you are granting these rights to the login under which you are currently logged on, then you must log off and back on in order for these newly-granted rights to take effect.

Install the TBSM components on the application server machine:

1. Insert the **TBSM Base Services** CD in the Application Server computer.
2. Double click the **Setup.exe** icon.
3. The System File Upgrade dialog opens. Read the text and click **Next**.
4. For the **Choose Destination Location** dialog, it is recommended that you place these files in a Temporary directory. Choose the **Browse** button to select your directory. Once a desired directory is selected, click **Next**.
5. When the **System File Upgrade** is complete, click **Finish**. If you are prompted to reboot the computer, do so now.
6. On the **Setup Type** dialog, choose **Custom** and click **Next**.

7. Follow instructions on the installation dialogs to continue the installation.
8. On the **Select Components** dialog, select the following components:

- **Application Server Files**
- **Tools and Utilities**
- **Propagation Agent Components**
- **Java Application Server**

Optionally, you may include the **Workstation Program Files** and **Help Files** components if you are going to use the TBSM client on the Application Server computer.

Note: If you are currently running TBSM 1.1.1, you also need to install **TBSM 1.1.1 Compatibility Services**. Do this by selecting **TBSM 1.1.1 Compatibility Services** on the **Select Components** dialog. This component enables TBSM 1.1.1 Workstations to communicate with the TBSM 1.5 server(s).

Select these 2 components:

- Tivoli BSM 1.1.1 Application Server
- Tivoli BSM 1.1.1 Notification Server

If the TBSM workstations and servers communicate through a firewall, the ports chosen for TBSM 1.5 should be different from the ports used previously in TBSM 1.1.1. **TBSM 1.1.1 Compatibility Services** should then be configured to use the TBSM 1.1.1 port settings and the new TBSM 1.5 services configured to use new port numbers.

After you migrate to TBSM 1.5, disable the TBSM 1.1.1 Compatibility Services (Tivoli BSM 1.1.1 Application Server and Tivoli BSM 1.1.1 Notification Server) from the control panel on the Application Server.

9. Click **Next**.
10. Enter the **hostname** of the local server machine. (By default, the hostname should be in the hostname field. If it is not, enter it.) Click **Next**.
11. Enter the **hostname** of the **SQL Server** machine, the SQL Administrator **Username**, and SQL Administrator **sa Password**. Click **Next**.
12. Enter the Web alias to be used by the Java Application server. **TBSM** is the recommended alias. Click **Next**.
13. Enter the **name** of the **Application Server** machine. Click **Next**.
14. On the **Select Program Folder** dialog, select the **Program Folder** to host all application files. Click **Next**.
15. On the **Setup Complete** dialog, when asked if you want to restart the computer, choose **No** and click **Finish**.

After installing the Application server, the following services should be on this computer:

- **Tivoli BSM Application Server**
- **Tivoli BSM Database Validator**

-
- Tivoli BSM Enqueue Proxy Server
 - Tivoli BSM Remote Execution Server
 - Tivoli BSM Console Server

Restarting the Servers

Reboot the SQL Database Server first. After the SQL Database Server is up and running, reboot the remaining servers in any order. **However, if you are using a SNA server, reboot it first after you reboot the SQL Database Server.** (For information about Installing and Configuring the Microsoft SNA Server, see Chapter 7 of this document.)

3

Installing TBSM/D Components

Information in this section explains how to install and configure components of TBSM for monitoring distributed systems only.

The base services for the TBSM - Distributed (TBSM/D) Edition should be installed on two Windows NT servers that we refer to as the Database Server and Application Server. SQL 7.0 is required on the Database Server but not on the Application Server. Additionally, each of the two servers requires that a different set of base services be installed on it.

Existing GEM Customers

Existing GEM Client or GEM Console customers should read and understand the material in this section because once this functionality is installed, TBSM replaces GEM Console. To install TBSM, the Tivoli Framework must be installed and running complete with a valid Tivoli Management Region (TMR) Server.

To enable Distributed Monitoring within TBSM:

1. Install event enablement on the Tivoli Enterprise Console (TEC) server, so events are forwarded to the TBSM Database Server.
2. Optionally, install the Tivoli Instrumentation Service on endpoints and managed nodes that will have Version 1.1 instrumentation deployed.

Before attempting to install TBSM - Distributed, refer to the Tivoli Business Systems Manager Release Notes, which contain the most up-to-date information for installing TBSM.

Installing TBSM - Distributed Edition on the TEC Server

Install Event Enablement, Task Server, and Java Runtime on the Tivoli Enterprise Console (TEC) so that events will be forwarded to the server.

If a previous version of Event Enablement is installed, uninstall it using the **tserver remove** command before installing TBSM – Distributed Edition on the TEC Server.

Install Event Enablement using one of the following methods:

- The standard Tivoli desktop

Follow the steps in Installing Components from the **Tivoli Desktop**. In **Step 3**, select the **Tivoli Business Systems Manager Event Enablement** component from the **Select Product to Install** scrolling list.

In Step 4, because you are installing EventEnablement on the TEC server, its machine name should be the only one listed in the **Clients to Install On list**. Move the other machine names from that list to the **Available Clients** list.

Repeat for the Tivoli Business Systems Manager JRE.

- The operating system command line

You can install event enablement from the command line using the `winstall` command as described in *Installing Components from the Command Line*, specifying `SERVER.IND` as the index file.

The following example shows a command you can use to install event enablement. If you are using Windows NT, you must run this command from the program shell.

winstall -c hostname:\$path -i SERVER.IND

For the JRE, issue: **winstall -c hostname:\$path -i JRE.IND**

When the installation is complete, you will see files in the following directories:

- For Intel: `%BINDIR%\TDS`
- For UNIX: `$BINDIR/TDS`

Installing Components from the Tivoli Desktop

Follow these steps to install TBSM - Distributed Edition from the Tivoli Desktop:

1. From the **Desktop** menu, select **Install-->Install Product** to display the **Install Product** dialog.

If the path to the TBSM - Distributed CD-ROM image has already been set, you should see the various TBSM - Distributed product components listed in the **Select Product to Install** list, similar to the following illustration.



2. If the TBSM - Distributed product components are not listed, click the **Select Media** button in the **Install Product** window to display the **File Browser** dialog.
 - The **File Browser** dialog enables you to specify the path to the installation media. Follow these steps if you know the path to the CD-ROM image:

-
- a. Enter the full path in the **Path Name** field.
 - b. Click **Set Path** to change to the specified directory.
 - c. Click **Set Media & Close** to save the new install media path and return to the **Install Product** dialog. The Install Product dialog now contains a list of TBSM - Distributed product components that are available for installation.

Follow these steps if you do not know the path to the CD-ROM image:

- a. From the **Hosts** list in the **File Browser** dialog, select the host that contains the installation media. When you select a host, the **Directories** list is updated to show the directories for the host you selected.
 - b. Select the directory that contains the installation media from the **Directories** list. The machine on which you are mounted should be in the same TMR as the machine on which you are installing.
 - c. Click **Set Media & Close** to save the new media path and return to the **Install Product** dialog. The dialog now contains a list of TBSM - Distributed product components that are available for installation.
3. From the **Install Product** dialog, select the desired product component from the **Select Product to Install** list.
 4. By default, all machines in the current TMR are listed in the **Clients to Install On** list in the **Install Product** dialog. This list should only contain machine names on which you want to install the desired component. Move the other machine names from that list to the **Available Clients** list by selecting one or more clients from the **Clients to Install On** list and clicking the right arrow button. The selected clients are moved from the **Clients to Install On** list to the **Available Clients** list.
 5. Click **Install** to install the desired component on the selected client machines. You can also click **Install and Close** to install the component and automatically close the **Install Product** dialog when installation is complete.
 6. The **Product Install** dialog is then displayed. It lists the operations that take place while the software is being installed and alerts you of problems you might want to correct

before installing the TBSM - Distributed product component.



7. Click **Continue Install** to begin the installation process, or click **Cancel** to stop the installation process. When the installation completes, the **Product Install** dialog displays a completion message.
8. Read the log in the product install dialog to check for errors.
9. Click **Close** when the Product Install status dialog indicates that the installation is complete. Repeat the above steps to select other TBSM - Distributed components to be installed on other sets of clients as desired.

Installing Components from the Command Line

Use the **winstall** command to install TBSM - Distributed product components from the operating system command line.

Note: Before you run any Tivoli command beginning with the letter w, you must set up the Tivoli Environment. A TMR server will usually have the environment set, but on other managed nodes, use one of the following:

- For NT: **setup_env.cmd**, located in the following subdirectory:
\winnt\system32\drivers\etc\tivoli
- For UNIX: **setup_env.sh**, located in the **/etc/tivoli** subdirectory

Following is the syntax for this command, followed by an explanation of its parameters:

```
winstall [-c cdrom_path] [-s tmr_server] [-i index_file] [-n] [-y]  
[install-variable=value...] [managed node...]
```

-c cdrom_path

Specifies the path of the CD-ROM image.

-s tmr_server

Specifies the name of the TMR server.

-i index_file

Specifies the index file from which the TBSM - Distributed component is to be installed. All index files have an extension of .IND. Following is a list of .IND files:

- Tivoli BSMEvent Enablement Version 1.1 (English): **SERVER.IND**
- Java Runtime Environment: **JRE.IND**
- Tivoli Instrumentation Service: **GEMISVC.IND**

-n

Specifies that the product is to be installed on all managed nodes that do not currently have the product installed. This argument is ignored if **managed-node** is specified.

-y

Specifies that the installation should proceed without confirmation. By default, this command identifies the actions that must be taken to perform installation and requests confirmation before continuing. Using this argument, **winstall** identifies the action and installs the product without requesting confirmation.

install-variable=value

A number of variables that control the installation can be set or defaulted on the command line. You can look in the install index file of the product for a definitive list of these variables. These variables specify required information or override default information.

managed node

Specifies the managed node on which a Tivoli Instrumentation Service will be installed. Multiple managed nodes can be specified, separated by a blank. It is important to specify the managed nodes because if nothing is specified, the product will be installed on all managed nodes in the TMR.

-c cdrom_path

Specifies the path of the CD-ROM image.

-s tmr_server

Specifies the name of the TMR server.

-i index_file

Specifies the index file from which the TBSM - Distributed component is to be installed. All index files have an extension of .IND. Following is a list of .IND files:

- Tivoli BSMEvent Enablement Version 1.1 (English): **SERVER.IND**
- Java Runtime Environment: **JRE.IND**
- Tivoli Instrumentation Service: **GEMISVC.IND**

-n

Specifies that the product is to be installed on all managed nodes that do not currently have the product installed. This argument is ignored if **managed-node** is specified.

-y

Specifies that the installation should proceed without confirmation. By default, this command identifies the actions that must be taken to perform installation and requests confirmation before continuing. Using this argument, **winstall** identifies the action and installs the product without requesting confirmation.

install-variable=value

A number of variables that control the installation can be set or defaulted on the

command line. You can look in the install index file of the product for a definitive list of these variables. These variables specify required information or override default information.

managed node

Specifies the managed node on which a Tivoli Instrumentation Service will be installed. Multiple managed nodes can be specified, separated by a blank. It is important to specify the managed nodes because if nothing is specified, the product will be installed on all managed nodes in the TMR.

Refer to the **winstall** command in the Tivoli Framework Reference Manual for more information.

Configuring the Connection between TBSM and the Tivoli Enterprise Console Server

The Tivoli Enterprise Console (TEC) Server is the primary collection point for the events that TBSM/D receives from its event sources. Bridging from the TBSM data server to the TEC environment requires the installation and the configuration of the TBSM Agent Listener and the TBSM Event Enablement components. Without properly installing and configuring these components, TBSM will not work as described.

Shipped as a Tivoli installable image, TBSM Event Enablement must be installed on each TEC server system where event forwarding to TBSM is required. The TBSM Event Enablement component also contains the TBSM Task Server component. Each is a long running process that listens on a TCP/IP port for TBSM/D connection requests. TBSM Event Enablement is responsible for forwarding TEC events to TBSM/D. The TBSM Task Server receives Tivoli task requests, executes the task on the TMR that it is on, and returns the output to TBSM/D.

As part of the process of enabling the TEC events to be forwarded to TBSM/D, TEC rules must be installed into the TEC rule engine to forward the events to one of the TBSM Event Enablement entry points. The **ihsttec.sh** utility is used to configure a TEC Rule Base with rules to forward APM events and DM events. The utility also installs the APM TEC event class file, `interapp.baroc`, so that APM events are recognized by the TEC engine. At times the **ihsttec.sh** utility may not be the best way to add a TEC forwarding rule.

See the *Tivoli Business Systems Manager Administration Guide* for details on starting, stopping, configuring, and maintaining the Tivoli TBSM Event Enablement components.

The TBSM Agent Listener is a Windows NT Service, installed with the base installation process of TBSM/D. After configuring it with a list of systems running the TBSM Event Enablement component, the Agent Listener opens a connection to each system, receives incoming events from Event Enablement, and initiates the processing of the TEC event. See the **gemeeconfig** command for details about how to define the list of Event Enablers to the Agent Listener.

In case the default listening ports used by TBSM Event Enablement and TBSM Task Server collide with other applications, each component allows its default listening port to be configured. If this action is taken be sure to configure the correct port on the TBSM side of the connection as well. The **gemeeconfig** and **tgmtaskconfig** commands are used to configure the TBSM client sides of the connections.

Installing Tivoli Instrumentation Service

If you're running an existing version of GEM Instrumentation Service (GEM IS or GEM IL), upgrading to Tivoli Instrumentation Service is optional.

The Tivoli Instrumentation Service is an interface between the applications in your business system and TBSM. It reports the status of your applications and their current state. This information is obtained from the application itself or from other sources, such as the operating system. If you are using instrumentation built with the Tivoli Module builder methodology, you must install the Tivoli Instrumentation Service. The Tivoli instrumentation Services acts as an interface for this type of instrumentation.

You can install the Tivoli Instrumentation Service and then use Tivoli Software Distribution to deploy it to managed systems and endpoints. The install creates a task library of commands and a Distributed Monitoring (DM) profile for managing and configuring the Tivoli Distributed Instrumentation Service, as well as a Software Distribution profile for distributing it to managed systems and endpoints. Install the Tivoli Instrumentation Service using one of the following:

Standard Tivoli desktop

Follow the steps in Installing Components from the Tivoli Desktop. In Step 3, select the **Tivoli Instrumentation Service** component from the **Select Product to Install** scrolling list, as well as the desired machines in your TMR from the **Client to Install On** list in step 4, moving all other machine names to the **Available Clients** list.

Operating system command line

You can install the Tivoli Instrumentation Service from the command line using the **winstall** command as described in Installing Components from the Command Line, specifying **GEMISVC.IND** as the index file. The following example shows a command you can use to install the Tivoli Instrumentation Service. If you are using Windows NT, you must run this command from the program shell:

```
winstall -c hostname:$path -i GEMISVC.IND -n -y
```

You can then distribute the Tivoli Instrumentation Service to the desired managed nodes and TMA endpoints using the following steps:

1. Open the **Applications_<region_name>** policy region.
2. Open the policy region for the Tivoli GEM application. This policy region is named: **PR_TIS_Default_<region_name>**
3. To subscribe endpoints, open the Profile Manager:
PM_Tivoli_Instrumentation_Service_1.1_Endpoint. Subscribe to this profile manager any TMA endpoints on which you want the TBSM - Distributed Instrumentation Service.
4. To subscribe managed nodes, open the Profile Manager:
PM_Tivoli_Instrumentation_Service_1.1 Subscribe to this profile manager any managed nodes on which you want the TBSM - Distributed Instrumentation Service.
5. Distribute the file package profile to subscribers as necessary. If you distribute the file package to the Profile Manager for endpoints (**PM_Tivoli_Instrumentation_Service_1.1_Endpoint**), the file package is distributed to all endpoints that are subscribed to that Profile Manager.

To distribute to specific subscribed endpoints, you must display the Distribute File Package window, expand the **PM_Tivoli_Instrumentation_Service_1.1_Endpoint** Profile Manager and select specific endpoints as distribution recipients.

Other Installation Options

After installing the TBSM event enablement, server, and console, you can install additional TBSM components depending on your environment. These components include the following:

- CICSplex SM Instrumentation
- DB2 for OS/390 Instrumentation
- Tivoli NetView for OS/390 application management interface (AMI). Refer to the *Tivoli NetView for OS/390 Customization Guide and the Tivoli NetView for OS/390 Installation and Administration Guide* for information on how to install the Tivoli NetView for OS/390 application management interface (AMI), system management event instrumentation, and topology subsystem instrumentation.
- Tivoli Manager products, such as MQSeries. Refer to the books shipped with the specific modules that you want to install.

Tivoli Business Systems Manager documentation is in PDF format and is available on the Tivoli Business Systems Manager Publications CD.

4

Configuring TBSM/D Edition

This chapter describes configuration tasks to perform before using TBSM - Distributed. Perform the general configuration steps for TBSM before configuring these components.

Configuring Event Enablement and Task Server

The following configuration items apply to both the task server and event enablement.

- Setting process level security
- Configuring TCP/IP ports
- Configuring when multiple Tivoli oserv processes are installed

Setting Security for Running the Task Server and Event Enablement

This section describes setting security for running the task server and event enablement on Windows NT and Windows 2000 platforms.

Note: UNIX does not require any additional security configuration.

You must add user rights to the user ID under which the task server and event enablement services run. User rights enable these services to perform the following functions:

- Act as part of the operating system
- Replace a process level token
- Log on as a service.

To set user rights:

Windows NT

1. Log on to an ID with administrator authority. You must set the user ID under which the Tivoli TBSM server runs as part of the Administrators group.
2. **Select Start > Programs > Administrative Tools > User Manager (or User Manager for Domains for NT servers).**
3. **Select Policies > User Rights.**
4. **Select Show Advanced User Rights.**
5. Scroll through the **Right:** list and select **Act as part of the operating system.**
6. The **Grant To:** list displays the groups and users currently granted this right. If your target user ID is already listed, go to step 10; otherwise, continue.
7. Click **Add.**

-
8. Click **Show Users**.
 9. Select the target user ID (the one under which the server will run) and click **Add**.
 10. Click **OK**.
 11. Repeat Step 5 through Step 9 for **Replace a process level token**.
 12. Repeat Step 5 through Step 9 for **Log on as a service**.
 13. Log off and log back on to the target ID for the user rights to take effect.

Windows 2000

1. Log on to an ID with administrator authority. You must set the user ID under which the Tivoli TBSM server runs as part of the Administrators group.
2. Select **Start->Settings->Control Panel->Administrative Tools->User Manager**.
3. On the tree, select **Local Policies->User Rights Assignment**.
4. Select **Act as part of the operating system** from the list of Rights, and ensure the login is in the **Grant To** list. Double-click on **Act as part of the operating system** to open it and add the login ID if needed.
5. Select **Replace a process level token** from the list of Rights.
6. Select **Log on as a service**.
7. Click **OK** and exit all windows.

Configuring TCP/IP Port Numbers

If you are using TCP/IP, the task server and event enablement default to using the following TCP/IP ports:

- Port 4020 for Tivoli NetView for OS/390 communications. This is optional if Tivoli NetView for OS/390 is part of the environment. Communications is between the task server and Tivoli NetView for OS/390.
- Port 4030 for event enablement, to agent listener communications.
- Port 4042 for task server communications.

No action is required if these port numbers do not conflict with your current application settings. However, if there is a conflict, change the port number or numbers using these instructions:

1. On the machine where the task server and event enablement are installed, open a command window.
2. Change to one of the following directories:
 - For NT: `%windir%\system32\drivers\etc`
 - For UNIX: `/etc`
3. Add the appropriate entry to the **services** file:

tserver_tbsm	4020/tcp	#server 390 port
tserver_ee	4030/tcp	#event enablement port
tserver_ts	4042/tcp	#task server port

Substitute your port number for the default port number or numbers listed above.

Notes:

1. If you reassign the 390 port, you must also customize the Tivoli NetView for OS/390 **Port** keyword in the **DUIFPMEM** file. Refer to the Tivoli NetView for OS/390 Administration Reference for more information on the **Port** keyword.
2. If you reassign the event enablement port, then you must also run the **gемееconfig** command on the agent listener's machine.
3. If you reassign the task server port, then you must also run **tgmtaskconfig** on the SQL Server machine.
4. For migration purposes, tserver_390 will continue to be recognized for the 390 port. However if NMC server is installed on the same machine as TBSM Event Enablement server, then the name used in the etc/services file for the 390 port should be changed to use tserver_tbsm rather than tserver_390 or a port conflict will occur.

Configuring for Network Address Translation

If a firewall exists between the Event Enablement machine and either OS/390 or the SQL server and network address translation (NAT) is being used, then **config -f** must be run to define the NAT address to the event enablement and task server processes. The address supplied to the **config -f** command is the address that the Event Enablement machine is known by outside of the firewall.

After running **config -f**, both the event enablement process and the task server process must be stopped, then both processes can be restarted. Stopping one process and restarting it, then stopping and restarting the other process will not allow this change to take affect. Both processes must be stopped and then restarted.

Configuring Multiple Tivoli Object Servers on the TEC Machine

The task server and event enablement scripts run the Tivoli **setup_env** script to ensure that the environment is correctly set up before starting any processes. The **setup_env** script is normally located in the following directory:

- For Windows NT: %windir%\system32\drivers\etc\Tivoli
- For UNIX: /etc/Tivoli

If you have multiple Tivoli object servers (oservs) installed on your machine, **setup_env** might not be located in the normal location. If this is the case for your installation, set the **TSERVER_ETC** environment variable to the directory that contains the appropriate **setup_env** file.

Configuring the Task Server

See these sections to help you configure the task server:

- Setting the user ID that TME tasks are run as
- Setting the user ID for OS/390 tasks
- Defining LU 6.2 configuration
- Configuring task server startup
- Enabling OS/390 command startup

For information on how to configure Tivoli NetView for OS/390 so that instrumentation events flow directly to the task server (without requiring Tivoli), refer to the *Tivoli NetView for OS/390 Customization Guide*.

Setting the User ID for Task Server Running TME Tasks

Tivoli Business Systems Manager supports TME Tasks. To run these tasks, the user ID that the task server is running under must be associated with an administrator on the TMR server.

On Windows NT, the user ID that the task server runs under can be defined by going into the **Control Panel**, selecting **Services**, selecting **TBSM Task Server**, and then selecting the **Startup** button. A Service panel for the task server opens. Under the **Log On As** section, select the radio button next to **This Account**, choose the user ID that the task server should run under, enter the password, and click **OK**. Restart the task server to pickup this change.

On the UNIX platforms, the task server runs as **root**.

To associate the user ID that the task server is running under with an administrator on the TMR server, follow these steps:

1. Determine the roles required to run all the tasks from the Tivoli desktop. To do this:
 - a. Open the Tivoli desktop.
 - b. Double-click **Policy Regions**.
 - c. Double-click **Task Libraries**.
 - d. Right-click each task icon. Select **Edit task**. Note the roles highlighted in the **Roles Required to Execute Task**. You need this information in subsequent steps. Right-click the selected administrator icon. Select **Edit Logins**.
 - e. Enter the login name you are using to run the GEM server.
 - f. Select **Change & Close**.
2. Select an administrator on the TMR server to associate with the user IDs that the task server is running under.
3. Right-click each task icon. From the pop-up menu, select **Edit Properties**.
4. Set the following values in the **Administrator Properties dialog**:
 - User Login Name: **\$root_user**
 - Group Name: **\$root_group**
 - Select **Change & Close**.
5. Right-click the selected administrator icon again. From the pop-up menu, select **Edit TMR Roles**.
6. Select the roles needed to run the task server tasks.
7. Select **Change & Close**.

Setting the User ID for OS/390 Tasks

The task server executes OS/390 commands associated with the GEM DB2 and CICS APM instrumentation under a single Tivoli NetView for OS/390 operator ID. To specify this operator ID, run the **tserver hostcmdoper** command to set the operator ID and password. This command encrypts the password and stores it on disk.

Defining the Logical Unit (LU) Name

The task server can communicate with Tivoli NetView for OS/390 using either TCP/IP or LU 6.2. To communicate with the Tivoli NetView for OS/390 environment using LU 6.2, you must define the LU name to be used for the server. For Windows NT or AIX, to define the LU name of the server, use the sample **ts.acg** configuration file in the following directory:

- For Intel: **%BINDIR%\TDS\server\sample**
- For UNIX: **\$BINDIR/TDS/server/sample**

A sample **ts.acg** file is shipped with TBSM - Distributed and represents a sample IBM Communications Server configuration file for task server LU 6.2 communications. This should only be used as a guide, as changes are most likely needed for your environment. Refer to the appropriate communications server documentation for more information on setting up LU 6.2 connections. The IBM Communications Server can be used on either Windows NT or AIX. Microsoft's SNA Server can also be used for LU 6.2 communications on Windows NT.

Starting Task Server Manually or Automatically

On NT platforms, the task server is automatically configured to start as a service when the system is booted. To modify the task server service, select **Start-->Settings-->Control Panel-->Services**. The name of the service is **Tivoli BSM Tasks Server**.

On UNIX platforms (AIX and Solaris), the task server process is set up to run as the daemon **ihscts**. The installation updates the appropriate system configuration files so that the process starts automatically when the machine is booted. The affected configuration files differ across platforms:

- Solaris: **/etc/rc3.d/S95tbsm** and **/etc/rc0.d/K05tbsm**
- AIX: **/etc/rc.tbsm**.

An entry has been added to **inittab** with the id **rctbsm**.

See **Starting Task Processor** and **Stopping Task Processor** for details on using the start and stop commands to start and stop the task processor manually.

Configuring the Task Server Properties File

The server properties file allows you to customize certain attributes of the Task Server. Refer to the file itself for information on the various attributes. The server properties file is located in the following directory:

- **\$BINDIR/TDS/EventService/config**

Enabling Task Server OS/390 Support

On the Tivoli NetView for OS/390 focal point, confirm you have the correct information on these two NetView members on the mainframe: **DUIFPMEM** and **DUIIGHB**

In **DUIFPMEM**, the TCPNAME parameter must match the step name used by the TCPIP task:

- USETCPIP parm must be **YES**.
- SOCKETS parm should be **50**.
- PORT parm number must match what you have in tserver_390 parm on the Task Server workstation.

In **DUIIGHB**, the TCPNAME parm must match the name used by the TCPIP task.

To start the connection between Tivoli NetView for OS/390 and the Task Server, execute the following command:

```
netconv action=start ip=hostnameWhereTaskServerIsRunning
```

An example of this command is:

```
netconv action=start ip=69.200.60.65
```

Configuring Event Enablement

After event enablement is installed, you can configure the workstation environment for use with instrumented business components in the following areas:

- Updating the TEC rules and event classes
- Setting the TME user ID
- Configuring event enablement startup
- Setting the maximum default size of the event cache

Updating TEC Rules and Event Classes

To update the TEC rules and event classes for event enablement, run the **ihsttec.sh** file, which is on the TEC server machine, in the following directory:

- For Intel: **%BINDIR%\TDS\EventService**
- For UNIX: **\$BINDIR/TDS/EventService**

This script sets up the rules base and event classes for TBSM Event Enablement events to pass through the TEC server. It also asks you to identify any Distributed Monitoring profiles you use for monitoring your operating systems.

Setting the User ID that Event Enablement Runs TME Tasks Under

The event enablement process must be capable of executing Tivoli Management Environment commands, therefore it must have sufficient authority within the Tivoli environment. To run these tasks, the user ID that event enablement is running under must be associated with an administrator on the TMR server.

On Windows NT, the user ID that event enablement runs under can be defined by choosing **Settings** from the Windows **Start** menu and then choosing **Control Panel-->Services**, and selecting **Tivoli BSM Event Enablement**, and then selecting the **Startup** button. A Service panel for the service opens. In the **Log On As** section, select the radio button next to **This Account**, choose the user ID that event enablement should run under, and enter the password. Click **OK**. Restart the event enablement service to pickup this change.

On the UNIX platforms, event enablement runs as **root**.

To associate the user ID that event enablement is running under with an administrator on the TMR server, follow these steps:

1. Determine the **roles** required to run all the tasks from the Tivoli desktop. To do this:
 - a. Open the Tivoli desktop.
 - b. Double-click **Policy Regions**.

- c. Double-click **Task Libraries**.
 - d. Right-click each task icon.
 - e. Select **Edit task**.
 - f. Note the roles highlighted in the **Roles Required to Execute Task**. You need this information in subsequent steps.
 - g. Right-click the selected administrator icon.
 - h. Select **Edit Logins**.
 - i. Enter the login name you are using to run the GEM server.
 - j. Select **Change & Close**.
2. Select an administrator on the TMR server to associate with the user ID that event enablement is running under.
 3. Right-click on the selected administrator icon. From the pop-up menu, select **Edit Properties**.
 4. Set the following values in the Administrator Properties dialog:
 - User Login Name: **\$root_user**
 - Group Name: **\$root_group**
 - Select **Change & Close**.
 5. Right-click the selected administrator icon again. From the pop-up menu, select **Edit TMR Roles**.
 6. Select the roles needed to run tasks.
 7. Select **Change & Close**.

Starting Event Enablement Manually or Automatically

On NT platforms, event enablement is automatically configured to start as a service when the system is booted. From the **Start** menu, select **Settings-->Control Panel-->Services** to modify the event enablement service. The name of the service is **Tivoli BSM Event Enablement**.

On UNIX platforms (AIX, Solaris, HP), the event enablement process is set up to run as the daemon **ihstdmai**. The installation updates the appropriate system configuration files so that the process starts automatically when the machine is booted.

The affected configuration files differ across platforms:

- Solaris: **/etc/rc3.d/S95ee_gem** and **/etc/rc0.d/K05ee_gem**
- AIX: **/etc/rc.eegem**
An entry has been added to **inittab** with the id **rceegem**.
- HP/UX: **/sbin/init.d/eegem** and **/etc/rc.config.d/eegem**
/sbin/init.d/eegem is linked to the following:
 - **/sbin/rc0.d/K110eegem**
 - **/sbin/rc1.d/K110eegem**
 - **/sbin/rc2.d/K110eegem**

Modify `/etc/rc.config.d/ee gem` to enable the daemon to start at bootup.

See *Starting Event Enablement Task Server* and *Stopping Event Enablement Task Server* in the *Tivoli Business Systems Manager Administration Guide* for details on using the `ee_start` and `ee_stop` commands to start and stop event enablement manually.

Setting the Maximum Default Event Cache Size

An event cache is used to hold the most recent events sent to event enablement. The maximum default size of the cache is 100 megabytes. You can specify the maximum amount of disk space that is used by the cache by editing the `ihsttec.cfg` file, which is on the TEC server machine, in the following directory:

- For Intel: `%BINDIR%\TDS\EventService\config`
- For UNIX: `$BINDIR/TDS/EventService/config`

Edit the following parameters in the `ihsttec.cfg` file:

```
CACHE_SIZE = size  
CACHE_ENABLED = value
```

where:

- size** Specifies the amount of disk space, in megabytes, to which the event cache is allowed to grow. If you specify 0, then the default cache size of 100 megabytes is used.
- value** Specifies whether or not the cache should be used. Specifying 0 turns off the cache. Any other value causes the cache to be used.

The location for the cache file is in the following subdirectory:

- For Intel: `%BINDIR%\TDS\EventService\db`
- For UNIX: `$BINDIR/TDS/EventService/db`

You must ensure that there is enough disk space available for the maximum cache size (100 megabytes).

Note: For these changes to take effect, restart Event Enablement.

Monitoring Tivoli Instrumentation Service

Tivoli TBSM Instrumentation Service is installed separately on your target systems. You may need to perform a few configuration steps to use certain functions, depending on whether your network configuration consists of the following:

- A standalone Tivoli Management Region (TMR) with its own local Tivoli Enterprise Console (TEC). In this case, no further configuration is needed. Managed nodes and endpoints in the TMR send their events to the local TEC.
- Interconnected TMRs in which only one contains the TEC. Managed nodes and endpoints, which do not have the TEC in their TMR, must send their events to the TEC in the interconnected, or non-local, TMR.
- Multiple TECs in interconnected TMRs. Additional configuration is required to specify the desired server to receive events.

Configuring Support for a Non-Local TEC

If your Tivoli network environment includes interconnected Tivoli Management Regions (TMRs) in which only one contains the TEC, you can still distribute TBSM instrumentation (built from version 2.3 script and Java templates) to Tivoli Management Agent (TMA) endpoints or managed nodes that do not have the TEC in their local TMR. Events generated from these systems are sent to the TEC in the interconnected TMR.

Configuring for TMA Endpoints

If your target system is a TMA endpoint sending events to a TEC in an interconnected TMR, you must define EventServer by running the following two commands in the local TMR:

- **wregister -ir EventServer**
- **wupdate -r EventServer All**

The only restriction in this case is that there must be only one TEC in the interconnected TMRs, since the server name is not being supplied. No additional configuration is necessary.

Configuring for Managed Nodes

If you require support for managed nodes as well as TMA endpoints, you must guarantee that the **wpostmsg** command is available on the managed node. The **wpostmsg** command is a platform-dependent executable, with accompanying run-time libraries, that is already provided on TMA endpoints. The **wpostmsg** command must exist in the oserv path, the TBSM - Distributed Instrumentation Service must be configured to utilize **wpostmsg**, and the EventServer must be defined as described in *Configuring Support for Multiple TECs*.

Configuring Support for Multiple TECs

If your interconnected TMRs contain multiple TECs, further configuration is required to identify the desired TEC server to receive events. Instrumentation generated using scripts or generated using TBSM - Distributed Instrumentation Service both send events to the local TEC using the mechanism appropriate for the management environment as follows:

- For managed nodes: **idlcalls**
- For TMA endpoints: **wpostmsg**

However, you can override this behavior or add support for non-local or multiple TECs by creating a file with the appropriate name as follows:

- For managed nodes: **tec_host.mn**
- For TMA endpoints: **tec_host.tma**

An example **tec_host.tma/mn** file follows:

```
#
#Note:
#the script template will only use the IDL, POSTMSG, and WPOSTMSG
#entries
#
IDL(1909366687.1.701#Tec::Server,liskay2)
POSTMSG(liskay2)
WPOSTMSG(@EventServer#liskay2-region,liskay2)
```

Each TMA or managed node file contains one or more records, and each record specifies an event server and the mechanism for conveying events to that event server. The logic for determining which event server(s) to send events to is as follows:

- Determine whether a **tec_host.mn** or **tec_host.tma** file exists in \$APPDIR, which points to the following directory:
 - For Intel: **%BINDIR%\..\generic_unix\TME\GEM **
 <gdf_manufacturer\<gdf_product\ <gdf_version
 \<cdf_manufacturer\<cdf_product\<cdf_version
 - For UNIX: **\$BINDIR/./generic_unix/TME/GEM/** <gdf_manufacturer/<gdf_product/
 <gdf_version /<cdf_manufacturer/<cdf_product/<cdf_version

If either file exists, use the event server(s) in that file.

- Determine whether a **tec_host.mn** or **tec_host.tma** file exists in the data directory for the environment. (\$DBDIR for managed nodes or \$LCF_DATDIR for TMA endpoints)
If either file exists, use the event server(s) in that file.
- If neither of the above locations has the appropriate **tec_host** file, use the default logic as follows:
 - For managed nodes: **idlcall**
 - For TMA endpoints: **wpostemsg**

Each line within a **tec_host.mn** or **tec_host.ma** file can consist of either a comment, beginning with a number sign (#) or any one of the following:

- **wpostemsg**(<eventserver, <eventserver hostname)
where

eventserver

Specifies the name of the event server in name registry format. The default is the local event server. To specify a remote server, enter **@EventServer#tmr**, where **tmr** is the name of a Tivoli Management Region (TMR) (for example, **@EventServer#spunky-region**).

eventserver hostname

Is the hostname where the specified event server resides.

See the *Tivoli Framework Reference Manual* for more information on the **wpostemsg** command.

- **postemsg**(<hostname)
where:

hostname

Specifies where the targeted event server resides.

- **idl**(<eventserver_OID, <hostname)
where:

eventserver_OID

Is the Tivoli ObjectID (OID) of the event server.

hostname

Specifies where the targeted event server resides

Notes:

1. **idl**(<eventserver_OID, <hostname) is not valid for a TMA endpoint.
2. If you specify an event server or multiple TEC event servers by using a **tec_host.mn** TMA file, the script or IS instrumentation will not send an event to the local TEC server, unless you specify the local event server in the **tec_host.mn** TMA file as well.

Configuring Monitoring of Tivoli Instrumentation Service

If you are running an existing version of GEM Instrumentation Service (GEM IS or GEM IL), upgrading to Tivoli Instrumentation Service is optional.

After installing TBSM - Distributed Instrumentation Service and distributing it out to the managed nodes and endpoints, you can configure heartbeat and state monitors to manage the status of the TBSM - Distributed Instrumentation Service itself, using the following steps:

1. Open the **Applications_<region_name** policy region.
2. Open the policy region for the following TBSM application:
PR_GEM_Default_<region_name. The policy region window is displayed.
3. Select the profile manager for the TBSM - Distributed Instrumentation Service and double-click on the icon to display the **Distributed Monitoring Profile Properties** dialog.
4. Use the buttons at the bottom of the dialog to add and delete monitors, edit monitor settings, and enable and disable the monitoring of TBSM - Distributed Instrumentation Service.

5

TBSM Enterprise Edition Components

This chapter describes the Tivoli Business Systems Manager (TBSM) Source/390 set of components that enable monitoring and management of OS/390 based systems. Information in this chapter is intended for the TBSM Enterprise Edition.

TBSM Source/390 resides on the OS/390 system and consists of the TBSM Source/390 Object Pump, Object Server, and Dataspace.

Installation of the Windows NT base components (services that are necessary for monitoring Source/390), is described in Chapter 2, Installing Tivoli Business Systems Manager Base Services. Base services components include the MVS Listener, MVS Event Handler, and MVS Sender services.

TBSM Source/390

TBSM Source/390 performs a variety of tasks and functions. The following table provides an overview of Source/390 functions that are available upon installation and initial configuration.

Function	Description
Console Monitoring	Collects information from the console, start, stop, or abend for registered objects, which can be Started Tasks, Batch Jobs, CICS Regions, DB2, and IMS.
OMEGAMON for MVS	Source/390 collects data from Candle's OMEGAMON/MVS and monitors exceptions, thresholds, tape waits, or anything available for OMEGAMON/MVS.
OMEGAMON for CICS	Collects data from Candle's OMEGAMON/CICS. Exceptions, thresholds, and problems are reported in Tivoli Business Systems Manager by iconic representation of the registered CICS Region, which resides on the operating system along with Source/390 and OMEGAMON/CICS resides. In addition, data is collected to provide CICS file availability for a registered CICS region.
OMEGAMON for DB2	Data from Candle's OMEGAMON/DB2 is collected and exceptions, thresholds and problems are reported in TBSM by icon representation of the registered DB2 region.
OMEGAMON for IMS	Data from Candle's OMEGAMON/IMS is collected and exceptions, thresholds and problems are reported in TBSM by icon representation of the registered IMS region
CICS TDQ Message File	CICS Region abends, CICS file availability and information about LU6.2 connections are extracted from the TDQ file.
External Data Interface (EDI)	Internal TBSM mechanism that allows messages to be posted to the Object Pump as though they were from a console.

Function	Description
OPS/MVS	EDI is used within client REXX code to pass data directly to the TBSM Source/390 Object Pump on the same MVS system.
Batch Schedules	Schedules from an automated job-scheduling package can be interpreted and automatically incorporated into Tivoli Business Systems Manager.
SA/390	Tivoli Business Systems Manager interfaces directly with SA/390 through the implementation of the EDI, which enables an external application to pass text-based messages directly into the TBSM Source/390 Object Pump on the same OS/390 system.
Tivoli Workload Scheduler (OPC)	Utilizing OPC exit 7, we are able to capture additions made to the current plan, made outside of the daily planning process. These updates can then applied to the baseline snapshot created during the initial discovery process (via the processing of the Daily Plan report). Exception WTOs are also trapped for OPC tasks and applied to the proper batch objects.
CICSplexSM	Collects availability information from CICSplex SM
DB2 & DB2PM	Collects availability information from DB2 Performance Monitor
RMF	Collects availability information from Resource Measurement Facility
IMS	Collects availability information for IMS

TBSM Source/390 Address Spaces

TBSM Source/390 consists of three address spaces on each MVS image to be monitored. These address spaces are:

Address Space	Description
TBSM Source/390 Object Pump	Performs data collection and traps events for monitored objects
TBSM Source/390 Dataspace	Acts as an intermediate repository for the event messages
TBSM Source/390 Object Server	Handles communications between Source/390 and the NT server

TBSM Source/390 Object Pump

The TBSM Source/390 Object Pump captures data from a variety of sources including the MVS system console, performance monitors, subsystems, CICS regions, and other applications and resources. Upon detecting an event, the TBSM Source/390 Object Pump formats a message according to the TBSM defined specification and forwards the formatted message to the TBSM Source/390 Object Server.

The TBSM Source/390 Object Pump provides the interface between the various sources of performance monitoring and event data and the Tivoli Business Systems Manager NT Application Server. It provides the functionality to allow the collection of messages and exceptions from the following sources:

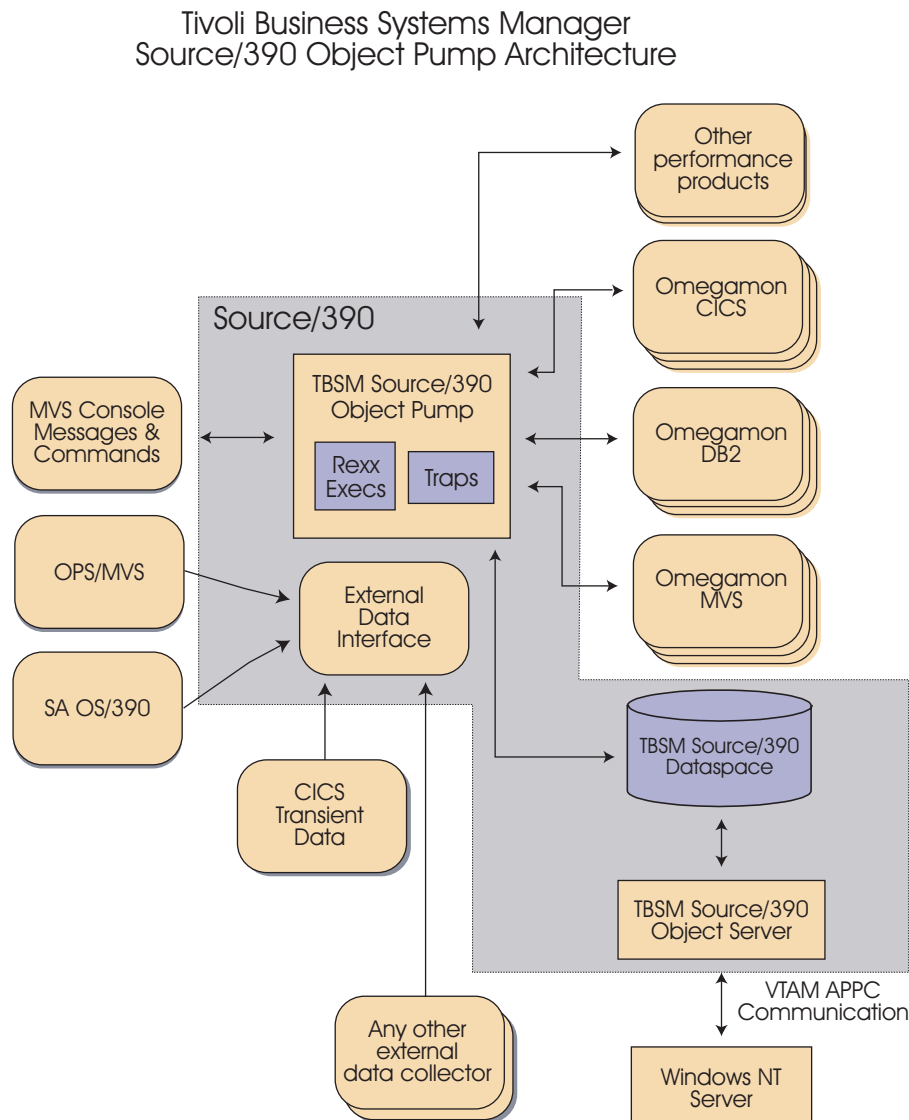
- The MVS console
- The Candle OMEGAMON series of products, in particular:
 - OMEGAMON for MVS
 - OMEGAMON for CICS
 - OMEGAMON for DB2

- OMEGAMON for IMS
- External sources of message data, for example:
 - OPS/MVS
 - CICS Transient Data Queues

Event and performance data collected by TBSM Source/390 Object Pump is stored in the TBSM Source/390 Dataspace on each MVS image. The TBSM Source/390 Object Server then picks up this data and exceptions and passes them to the Windows NT Server using APPC LU6.2 or IP communications for subsequent processing and storing in the Tivoli Business Systems Manager event database.

TBSM Source/390 Object Pump Architecture

The following graphic illustrates the TBSM Source/390 Object Pump and its data sources. It also shows the components that make up TBSM Source/390 and the passing of data from the TBSM Source/390 Object Pump to the TBSM Source/390 Dataspace, then to the TBSM Source/390 Object Server, and finally to the Windows NT-based Server.



TBSM Source/390 Object Server

The TBSM Source/390 Object Server provides communications between the TBSM Source/390 Object Pump and TBSM Source/390 Application Server. It passes the data gathered by the TBSM Source/390 Object Pump to the TBSM Source/390 Application Server, logs any activity, and manages communications with the TBSM Source/390 Application Server. The TBSM Source/390 Object Server communicates with the TBSM NT Server using LU6.2 APPC or IP connectivity. The TBSM Source/390 Object Server's primary task is routing and controlling the flow of event notifications using queue management.

TBSM Source/390 Dataspace

The TBSM Source/390 Dataspace is created to provide an intermediate queue area that can be accessed by the TBSM Source/390 Object Pump and the TBSM Source/390 Object Server to pass data and information between the two address spaces.

6

Installing & Configuring TBSM Source/390

This chapter describes how to configure TBSM Source/390 (mainframe components).

VTAM Definitions

The following definition is required only when using SNA communications to TBSM NT. The definition is required by Source/390 on each MVS image on which it is to execute.

APPL Definitions

name	VBUILD	TYPE=APPL
WTSKappl	APPL	ACBNAME=WTSKappl, MODETAB=modetab, AUTH=(ACQ), APPC=YES, AUTOSSES=2, DDRAINL=ALLOW, DMINWNL=1, DMINWNR=1, DRESPL=ALLOW, DSESLIM=4, PARSESS=YES, DLOGMOD=LU62PS, VPACING=0

Ensure that this applid has access to the Host Integration Server LU (e.g. add any CDRSCs needed or update SME exit as required). The applid and the LU should both be able to initiate sessions with one another.

WTSKappl is the VTAM applid defined for the Object Server on this MVS image.

Mode Table

The following definition is required only when using SNA communications to TBSM NT. TBSM LU 6.2 communications require a mode table containing the LU62PS and SNAVCMG entries. This mode table should be named on the MODETAB=modetab parameter of the APPL statements required for TBSM. The following shows an example of a mode table for use with Source/390, which can be assembled and placed in your VTAM library. Please note that the RUSIZES parameter in the mode table represents a maximum RU size of 1024 bytes. This parameter is also configured on the Microsoft Host Integration Server and must match the mode table otherwise unpredictable results may occur.

Name	MODETAB
MODEENT	LOGMODE=LU62PS, FMPROF=X'13',TSPROF=X'07', PRIPROT=X'B0',SECROT=X'B0',COMPROT=X'D0B1', RUSIZES=X'8787',ENCR=B'0000',TYPE=0,

```
MODEENT          LOGMODE=SNASVCMG,
  FMPROF=X'13',TSPROF=X'07',
  PRIPROT=X'B0',SECPROT=X'B0',COMPROT=X'D0B1',
  RUSIZES=X'9797',ENCR=B'0000',TYPE=0,
  PSERVIC=X'060200000000000000002300',
  APPCNOS=SNASVCMG
```

LOGMODE Entries for 3270 Pool Terminals

The TBSM Source/390 Object Pump requires a VTAM logmode name for each terminal model type that is supported (The TBSM Source/390 Object Pump can connect to a VTAM application as a model 2, 3, 4 or 5 terminal). By default, it uses the following LOGMODE names that are defined in the default VTAM LOGMODE table, ISTINCLM:

```
*****
*
*          3274 MODEL 1A WITH MODEL 2 SCREEN (LOCAL SNA)
*          PRIMARY SCREEN 24 X 80 (1920)
*          NO ALTERNATE SCREEN DEFINED
*
*****
D4A32782 MODEENT LOGMODE=D4A32782,FMPROF=X'03',TSPROF=X'03',PRIPROT=X'B*
          1',SECPROT=X'90',COMPROT=X'3080',RUSIZES=X'87C7',PSERVIC*
          =X'020000000000185000007E00',APPNCOS=#CONNECT
*****
*          3274 MODEL 1A WITH MODEL 3 SCREEN (LOCAL SNA)
*          PRIMARY SCREEN 24 X 80 (1920)
*          ALTERNATE SCREEN 32 X 80 (2560)
*
*****
D4A32783 MODEENT LOGMODE=D4A32783,FMPROF=X'03',TSPROF=X'03',PRIPROT=X'B*
          1',SECPROT=X'90',COMPROT=X'3080',RUSIZES=X'87C7',PSERVIC *
          =X'020000000000185020507F00',APPNCOS=#CONNECT
*****
*          3274 MODEL 1A WITH MODEL 4 SCREEN (LOCAL SNA)
*          PRIMARY SCREEN 24 X 80 (1920)
*          ALTERNATE SCREEN 43 X 80 (3440)
*
*****
D4A32784 MODEENT LOGMODE=D4A32784,FMPROF=X'03',TSPROF=X'03',PRIPROT=X'B*
          1',SECPROT=X'90',COMPROT=X'3080',RUSIZES=X'87C7',PSERVIC*
          =X'02000000000018502B507F00',APPNCOS=#CONNECT
*****
*          3274 MODEL 1A WITH MODEL 5 SCREEN (LOCAL SNA)
*          PRIMARY SCREEN 24 X 80 (1920)
*          ALTERNATE SCREEN 27 X 132 (3564)
*
*****
D4A32785 MODEENT LOGMODE=D4A32785,FMPROF=X'03',TSPROF=X'03',PRIPROT=X'B*
          1',SECPROT=X'90',COMPROT=X'3080',RUSIZES=X'87C7',PSERVIC*
          =X'02000000000018501B847F00',APPNCOS=#CONNECT
*****
```

Tivoli NetView Customization

This section is an overview of the customization needed for the TBSM Feeds that run on NetView/390.

For customization details, please refer to the following TBSM manuals:

- TBSM DB2 Release Notes
- TBSM CICSplex SM Release Notes
- TBSM IMS Release Notes
- TBSM MainView for DB2 Release Notes
- TBSM MainView for CICS Release Notes
- TBSM MainView for IMS Release Notes

-
- TBSM MainView for OS/390 Release Notes
 - TBSM TMON for DB2 Release Notes
 - TBSM TMON for CICS Release Notes
 - TBSM TMON for MVS Release Notes

NetView Started Task JCL

The following datasets should be added to the indicated concatenation within the Tivoli NetView Start Task JCL:

STEPLIB

hlq.SGTMMODS	TBSM modules
hlq.SEAGALT	Rexx support
hlq.SDSNLOAD	DB2 Loadlib

DSICLD

hlq.SGTMEEXEC	TBSM Clists
---------------	-------------

DSIPARM

hlq.SGTMSAMP	TBSM Sample files
--------------	-------------------

DSIMSG

hlq.SGTMMSGS	TBSM Message files
--------------	--------------------

NetView DSIPARM Member Customization

DSITBL01 (or whatever the active AUTOTBL name is)

%INCLUDE IHSBMAT	DB2
%INCLUDE IHSCMAT1	CICS
%INCLUDE IHSIMAT	IMS
%INCLUDE IHSBMVAT	MainView DB2
%INCLUDE IHSCMVAT	MainView CICS
%INCLUDE IHSIMVAT	MainView IMS
%INCLUDE IHSZMVMA	MainView OS390
%INCLUDE IHSBTMAT	TMON DB2
%INCLUDE IHSCTMAT	TMON CICS
%INCLUDE IHSZTMAT	TMON MVS

DSICMDU

%INCLUDE GTMCMD	TBSM Base and CICS
%INCLUDE IHSBCMD	DB2
%INCLUDE IHSICMD	IMS

DSIOPFU

%INCLUDE IHSBOPR	DB2
%INCLUDE IHSIMOPR	IMS
%INCLUDE IHSBMVOP	MainView DB2
%INCLUDE IHSCMVOP	MainView CICS
%INCLUDE IHSIMVOP	MainView IMS
%INCLUDE IHSZMVOP	MainView OS390
%INCLUDE IHSBTMOP	TMON DB2
%INCLUDE IHSCTMOP	TMON CICS
%INCLUDE IHSZTMOP	TMON MVS

DSIDMNB

CNMTAMEL to have DUIISFP on the Focal Point NetView.

MVS Performance

The Source/390 address spaces are automatically marked non-swappable via the SYSEVENT macro within the code. Do not mark the programs NON-SWAPPABLE via the OS/390 Program Properties table. The address spaces should however be given a dispatching priority similar to an online CICS Region. If Source/390 does not have adequate performance levels, communication problems may occur.

Running Multiple Copies of Source/390 on an MVS Image

To run multiple copies of Source/390 on an MVS image (to test service for example while the current version is still running) you must add an additional DD card to each of the three address space's startup JCL. The DD card should be coded as follows:

```
//ACC1Idxx DD DUMMY
```

Where:

xx can be any two characters valid for a DD name. When this DD card is omitted, the ID used by Source/390 defaults to '01.' Therefore when running only one copy of Source/390 on an MVS image, this DD card is not required.

JES3 Support

In a JES3 environment, started tasks console (STC) messages are the same as batch job console messages. Because of this, TBSM cannot distinguish between a message from an STC and one from a batch job. If a TBSM administrator adds batch jobs and STCs to an operating system with identical names under JES3, unpredictable results will occur.

Refer to the appropriate Release Notes for TBSM functions you want to implement for information about other JES3 limitations.

TBSM Startup Parameter Considerations

This section discusses continuing parameters and system variables for TBSM Object Pump and TBSM Object Server.

Note: An asterisk (*) in Column 1 indicates a comment.

■ Continuing Parameters

If a parameter needs to be continued onto one or more cards, then specify a comma followed by a space at the end of the data and start the continued data in Column One of the next card.

Thus:

```
KEYWORD=partof,
thedata
```

is the same as:

```
KEYWORD=partofthedata
```

Notice that the comma is not part of the data for the keyword. If you want a comma to be part of the data, code two commas. Either of the following is valid:

```
KEYWORD=part1,,
part2
```

or
KEYWORD='part1',
'part2'

Both of these are the same as

KEYWORD=part1,part2

The data for KEYWORD is part1,part2.

The first comma is included in the data; the last comma indicates that the data is continued on the next card.

■ **System Variables**

Within the Source/390 Object Pump parameter input stream, you may use system variables within the parameter data; for example &SYSNAME. When using system variables, you must code a trailing period after the variable name; for example CONSOLE_NAME=&SYSNAME.

You may also code the two characters ‘..’ (dot dot) or the variable name “&ID.” or “&ACC1ID.” anywhere within the parameter data. These will be replaced by the current ID characters from the ACC1Idxx DD card (where xx are the ID characters) in the Source/390 Object Pump startup JCL.

TBSM Source/390 Dataspace Startup Parameters

The following parameters should be placed in the hlq.SGTMSAMP member referenced by the Source/390 Dataspace startup JCL:

- DATASPACE SIZE=dataspace size
- OSQUEUE SIZE=osqueue size
- OPQUEUE SIZE=opqueue size

Where:

dataspace size is the size of the TBSM Source/390 Dataspace in bytes. The data space is sub allocated into cells. At least one cell is required to hold each message. A dataspace size of 10000000 can hold approximately 35,000 records.

osqueue size is the size of the TBSM Source/390 Object Server queue in pages (4096 bytes). The queue is used to send data between the TBSM source/390 object pump and the TBSM source/390 object server. One page can hold approximately 1000 records. It is recommended that none of the queues exceed the capacity of the dataspace.

opqueue size is the size of the TBSM Source/390 Object Pump queue in pages (4096 bytes). The queue is used to send data between the TBSM source/390 object server and the TBSM source/390 object pump. One page can hold approximately 1000 records. It is recommended that none of the queues exceed the capacity of the dataspace.

TBSM Source/390 Object Server Startup Parameters

The following parameters should be placed in the hlq.SGTMSAMP member referenced by the TBSM Source/390 Object Server startup JCL. There are two members in hlq.SGTMSAMP that can be used with the GTMSRVR PROC. PARMSRIP contains a sample set of IP control cards. PARMSRVR contains a sample set of LU6.2 control cards.

```
*
* Comments start with an asterisk in column one
*
PROTOCOL=LU62 | IP
REMOTE_APPLID=remote_applid
LOCAL_LUNAME=local_luname
LOGMODE=LU62PS | logmode
TCPIP_ADDRESS=tcip_address
*TCPIP_NAME=tcip_name
TCPIP_PORT=port_number
TCPIP_SERVER_ADDRESS=tcip_server_address
*TCPIP_SERVER_NAME=tcip_server_name
TCPIP_SERVER_PORT=port_server_number
TCPIP_JOBNAME=tcip_jobname
TIMEOUT=30 | timeout_value
CODEPAGE=037 | codepage
BUFFERSIZE=1 | buffersize
WSQUEUESIZE=wsqueuesize
TCPIP_ADDRESS_BACKUP=tcip_backup_address
*TCPIP_NAME_BACKUP=tcip_backup_name
TCPIP_PORT_BACKUP=tcip_backup_port
RETRY_INTERVAL=30 | retry_interval
RETRY_COUNT=0 | retry_count
VALIDCLIENT=client_ip_address | client_hostname
LOG1=log_dataset1
LOG2=log_dataset2
REGTRACE=NO | YES
```

Where:

PROTOCOL=LU62 | IP specifies the protocol that is used for the communication between TBSM source/390 and TBSM running on NT. The default is LU62.

LU6.2 Interface

When using the LU6.2 interface, the following control cards are required:

REMOTE_APPLID=remote_applid specifies the applid used by Object Server. The applid name must match the remote applid used in the Microsoft Host Integration Server.

LOCAL_LUNAME=local_luname specifies the luname used by the SNA Client. The Local luname must match the local luname used in the Microsoft Host Integration Server.

LOGMODE=LU62PS | logmode specifies the logmode that is used to define the parameters for the LU6.2 connection.

TCP/IP Interface

The following control cards are used for the TCP/IP interface.

TCPIP_ADDRESS=nnn.nnn.nnn.nnn
TCPIP_NAME=hostname

Specifies the IP address or hostname for the connection that is used to communicate with the ASIMVSIPOSLListenerSvc service. If the TCPIP_NAME parameter is used, object server will

attempt Host Name Resolution to locate the IP address. These are required operands. However, only one of these parameters is required to be specified.

The nnn operand of the ADDRESS parameter is to be separated by periods '.' and can be any number in the range from 1 to 255.

The computer_name is alphanumeric and the length can range from 1 to 255 characters long.

If **traceresolver** is set on in the TCPIP parameters, TCPIP debug messages are issued to the job log when host name resolution is attempted. These messages are issued whether or not the IP Address is found.

The **resolvertimeout** and **resolverudpretries** parameters of TCP/IP control the time specifications for Host Name Resolution processing within the TCP/IP network.

Refer to the TCP/IP manuals for more information regarding the TCP/IP parameters described here.

Note: If the IP address of the TBSM listener is setup in the NAT configuration, the NAT IP address must be used. If the NAT IP address is not used, the Object server will not connect to TBSM.

TCPIP_PORT=nnnnn

Specifies the port number to be used on behalf of the connection. nnnnn can be in the range from 1 to 32767. This is a required operand.

TCPIP_SERVER_ADDRESS=nnn.nnn.nnn.nnn **TCPIP_SERVER_NAME=hostname**

Specifies the IP address or hostname for the machine that object server is running on. The IP address is obtained dynamically if these cards are not specified. If the TCPIP_NAME parameter is used, object server will attempt Host Name Resolution to locate the IP address. These are required operands. However, only one of these parameters needs to be specified.

The nnn operand of the ADDRESS parameter is to be separated by periods '.' and can be any number in the range from 1 to 255.

The hostname is alphanumeric and the length can range from 1 to 255 characters long.

If **traceresolver** is set on in the TCP/IP parameters, TCP/IP debug messages are issued to the job log when host name resolution is attempted. These messages are issued whether or not the IP Address is found.

The **resolvertimeout** and **resolverudpretries** parameters of TCP/IP control the time specifications for Host Name Resolution processing within the TCP/IP network.

Refer to the TCP/IP manuals for more information regarding the TCP/IP parameters described here.

TCPIP_SERVER_PORT=nnnnn

Specifies the server port number to be used for the IP listener that is used to receive registration data from TBSM. This is a required operand.

VALIDCLIENT=nnn.nnn.nnn.nnn | client_hostname

Specifies an IP address or hostname of a TBSM client that is authorized to send data to object server. This IP address or hostname is the same resource as used on the TCPIP_ADDRESS or TCPIP_NAME card. This is a required operand.

When specifying an address, the nnn operand of the ADDRESS parameter is to be separated by periods '.' and can be any number in the range from 1 to 255.

The client hostname is alphanumeric and the length can range from 1 to 255 characters long.

VALIDCLIENT_HOSTRES=YES | NO

Specify NO to disable the VALIDCLIENT function from attempting to resolve the hostname specified on the VALIDCLIENT card. The default is YES.

Note: Refer to the section titled, *Installing Event Handler (Configuration 3) Using IP* in Chapter 7 for information about installing TBSM to use IP communication services.

Optional Parameters for TCP/IP Interface

TCPIP_JOBNAME=tcPIP_jobname

Specifies the TCP/IP address space that is to be used to provide TCP/IP services. When not specified, the TCP/IP address space is dynamically retrieved. The TCP/IP Jobname cannot exceed 8 characters in length. This is an optional operand.

TIMEOUT=30 | seconds

Specifies the timeout value in seconds in which the TCPIP services will terminate if there is no response. The default is 30 seconds. The maximum value is 86400 seconds. This is an optional operand.

This parameter works in conjunction with the TCP/IP **resolvertimeout** parameter. For example, when the value of this parameter is less than the **resolvertimeout** parameter, the IP service requests will time out prior to TCPIP terminating the outstanding service request.

Refer to the TCP/IP manuals for more information regarding the **resolvertimeout** TCP/IP parameter.

CODEPAGE=037 | codepage

Specifies the codepage number to be used by the **ASIMVSIPOSListenerSvc** to convert the data to the appropriate ASCII symbols. This is an optional control card. The default is US codepage 037.

BUFFERSIZE=1 | size

Specifies the size (X1000 in bytes) of the Buffer used for sending and receiving data over the TCPIP Connection. The default is 1 (1000 bytes). This is an optional parameter. Size can be in the range from 1 (1000 bytes) to 32 (32000 bytes).

The following backup parameters are optional and are only to be used in the event that recovery is desired.

TCPIP_ADDRESS_BACKUP=nnn.nnn.nnn.nnn

TCPIP_NAME_BACKUP=hostname

Specifies the IP address or hostname for a backup connection that is to be used to communicate with the ASIMVSIPOSListenerSvc service in the event the primary connection fails. If the TCPIP_NAME_BACKUP parameter is used, object server will attempt Host Name Resolution to locate the IP address. These are optional operands. However, only one of these parameters is required to be specified for the backup resource.

The nnn operand of the ADDRESS parameter is to be separated by periods '.' and can be any number in the range from 1 to 255.

The computer_name is alphanumeric and the length can range from 1 to 255 characters long.

If **traceresolver** is set on in the TCPIP parameters, TCP/IP debug messages are issued to the job log when host name resolution is attempted. These messages are issued regardless of whether the IP Address is found.

The **resolvertimeout** and **resolverudpretries** parameters of TCPIP control the time specifications for Host Name Resolution processing within the TCPIP network.

Refer to the TCP/IP manuals for more information regarding the TCP/IP parameters described here.

Note: If the IP address of the TBSM listener is setup in the NAT configuration, the NAT IP address must be used. If the NAT IP address is not used, the Object server will not connect to TBSM.

TCPIP_PORT_BACKUP=nnnnn

Specifies the backup port number to be used on behalf of the connection. nnnnn can be in the range from 1 to 32767. This is a required operand.

RETRY_INTERVAL=30 | seconds

Specifies the retry interval in seconds that a connection is attempted to TBSM. This operand applies to both the primary or backup IP resource. The default is 30 seconds. This is an optional operand.

RETRY_COUNT=0 | retry_count

Specifies the number of retry attempts before switching to the alternate IP resource specified on the TCPIP parameters. The default is 0 indicating that automatic switching will not occur in the event the connection fails. The maximum value that may be specified is 999999.

Common Parameters for Both LU6.2 and IP Interfaces

LOG1=log_dataset1

LOG2=log_dataset2

Specifies the LOG datasets used to receive the TBSM events for event tracking and diagnostic purposes. These are required parameters.

TRACE=YES | NO

YES specifies that the debug messages are to be written to the console. The default is NO. This is an optional operand.

REGTRACE=YES | NO

YES specifies that the object registration messages received from TBSM are written to AOPLOG. The default is NO. This is an optional operand.

WSQUEUESIZE=wsqueuesize

wsqueuesize is the size of the TBSM source/390 work station queue in pages (4096 bytes). The queue is used to send data between the TBSM source/390 object server and the TBSM NT components. One page can hold approximately 1000 records. It is recommended that none of the queues exceed the capacity of the dataspace. This queue will begin to fill up when communication is lost between the object server and the NT components.

Object Server Problem Analysis

This section covers object server problem analysis. section

Connectivity Problems

If you have a problem with connectivity, contact the TCP/IP administrator. To determine if there is connectivity, use the PING command by IP address and Hostname between OS/390 and the TBSM Event Handler server from both directions.

Event Tracing

Use the following DDNAMES as a switch to enable additional tracing of the TBSM events during the transmission of data to TBSM.

TRACE DD coded in the JCL requests that the data records transmitted to TBSM be written to the specified file. The lengths transmitted to TBSM are visible, as the length field precedes the TBSM event message.

LOG DD coded in the JCL requests that logging information be written to the file specified on this ddname.

ASIMVSIPOSListenerSvc NT

The TRACE and LOG files should be used in conjunction with the ASIMVSIPOSListenerSvc running with Log Level 0. The Tivoli Support Center may require documentation from these three sources:

- Output from the TRACE File generated by Object Server
- Output from the LOG File generated by Object Server
- The LOG file generated by the ASIMVSIPOSListenerSvc running TBSM on NT

Source/390 Security

A userid must be assigned to Source/390 started tasks upon startup. The assigned userid(s) must have the necessary authority to update the Source/390 log datasets and read all other datasets coded in the Source/390 JCL. In addition to this authority, the Object Pump also requires OPERCMDS and OMVS security definitions.

OPERCMDs Security Definition

If the RACF class OPERCMDS is active, the resource MVS.MONITOR must be defined and the TBSM Source/390 Object Pump address space must have read access to this resource. Use the following command to achieve these requirements:

```
PERMIT MVS.MONITOR CLASS(OPERCMDs) ID(pump_userid) ACCESS(READ)
```

Where *pump_userid* is the RACF userid assigned to the PUMP address space.

Defining an OMVS Segment for TBSM Source/390 Object Pump

The user ID associated with TBSM Source/390 Object Pump requires that an OMVS segment be defined in Resource Access Control Facility (RACF).

Note: The tasks, examples, and references in this section assume that you are using the CS for OS/390 Security Server (RACF). If you are using a security product from another vendor, read the documentation for that product for instructions on task performance.

One way of satisfying the requirement for an OMVS segment in RACF is to define an OMVS RACF segment for the associated user ID. An easier way of accomplishing this task is to take advantage of the default OMVS segment support provided by RACF and OS/390 UNIX for users and groups.

Notes:

1. An HFS must be defined for the OMVS segment, and the home directory must exist.
2. If you use a trusted or privileged started task in ICHRIN03 or the STARTED class (especially a generic entry), be careful in assigning a default User ID (UID) and Group ID (GID) with facility class BPX.DEFAULT.USER. Whenever trusted or privileged is specified, all default tasks have superuser authority.

The default OMVS segments reside in the USER profile and GROUP profile. The names of these profiles are identified by the installation, using the BPX.DEFAULT.USER facility class profile. The application data field in the class profile contains the user ID, or the user ID/group ID, that is used to access the default OMVS segments for users and groups, respectively.

To set up default OMVS segments, follow these steps:

1. Define a Group ID (GID) to the system to be used as an anchor for a default OMVS group segment. Use the following command: **ADDGROUP OEDFLTG OMVS(GID(777777))** It is recommended that you make the **GID** unique so that it is easily identifiable. The **GID** can be either very high or very low. The other fields related to the **GID** are not likely to be used for anything.
2. Define a user ID (UID) to be used as an anchor for the default OMVS user segment. Use the following commands: **ADDUSER OEDFLTU DFLTGRP(OEDFLTG) NAME('OE DEFAULT USER') OMVS(UID(999999) HOME('/') PROGRAM('/bin/sh'))**

Note: To avoid giving superuser authority, do not use zero as the UID.

When defining a **UID**, consider the following:

- **UID** - Make the UID unique so that it is easily identifiable. The number can be very high or very low.
- **HOME** - Choose one of the following options when defining the home directory for the default user:
 - Define the HOME directory as the root (/). The users do not have write access, but do not need to update their home directory.
 - Define the HOME directory in the /tmp directory.
 - (Not recommended) Define a directory as you would for any other user. This directory is then used concurrently by many users that do not have an OMVS segment.
- **PROGRAM** - Define the default shell in this field.

The other fields related to this UID are not likely to be used.

3. Set up a default for the USER OMVS segment or set up a default UID and GID:

- To set up a default for the USER OMVS segment only, create a facility class profile named BPX.DEFAULT.USER, and then specify the default UID in the application data field. Use the following commands:
 - **RDEFINE FACILITY BPX.DEFAULT.USER APPLDATA('OEDFLTU')**
 - **SETROPTS RACLIST(FACILITY) REFRESH**

Note: You cannot set up a default GROUP OMVS segment alone.

- To set up a default UID and GID, create a facility class profile named BPX.DEFAULT.USER, and then specify the default UID and GID in the application data field. Use the following commands:
 - **RDEFINE FACILITY BPX.DEFAULT.USER APPLDATA('OEDFLTU/OEDFLTG')**
 - **SETROPTS RACLIST(FACILITY) REFRESH**

Be aware that the facility class must be activated. In addition, the USER profile of the default UID and the GROUP profile of the default GID must exist, and must contain OMVS segment with a UID and GID, respectively.

Note: RACF does not check to ensure that the application data points to a valid UID or UID and GID, or that the USER and GROUP profiles contain OMVS segments with the required UID and GID.

The following process shows how the BPX.DEFAULT.USER facility class profile works:

1. A user requests a UNIX service, which is serviced by the kernel.
2. The kernel calls the security product to extract the UID, GID, HOME, and PROGRAM information.

-
3. The security product attempts to extract the OMVS segment associated with the user. If the user is not defined, the security product attempts to extract and use the OMVS segment for the default user that was listed in the BPX.DEFAULT.USER profile.

A similar process is followed to obtain a GID when the user default group does not have an OMVS segment.

TBSM Source/390 Object Pump Startup Parameters

■ Parameters

Place the following parameters in the hlq.SGTMSAMP member referenced by the Source/390 Object Pump startup JCL:

```
*
* Comments start with an asterisk in Column One
*
SUBSYSTEM_NAME=TM.. | name | NO
CONSOLE=TM390.. | name
CONSOLE_MSCOPE=LOCAL | ALL
MAX_TRAPS=2016 | number
TERMINAL_PREFIX=TM390 | prefix
NUMBER_OF_TERMINALS=0 | n
MODEL2_LOGMODE=D4A32782 | name
MODEL3_LOGMODE=D4A32783 | name
MODEL4_LOGMODE=D4A32784 | name
MODEL5_LOGMODE=D4A32785 | name
LOG_SCREEN=NO | YES
EDI_BUFFER_SIZE=2048 | size
EXEC_REFRESH=YES | NO
SUBSYSTEM_MAXTHREADS=2 | threads
MAXTHREADS_PROMPT=YES | NO
*
OMEGAMON_REFRESH_PERIOD=00:04:00 | hh:mm:ss
OMEGAMON_EXCEPTION_CNTRL=ALL | DELTA
TRAP_REPORT=SYSOUT(A) | sysout(class) | dsname
LOG_FILE=SYSOUT(A) | sysout(class) | dsname
LOG_FILE_RECORD_LENGTH=200 | length
LOG_SPIN_INTERVAL=04:00:00 | hh:mm:ss
TRACE=YES | NO
COMMAND_PREFIX=subsystem_name | prefix
OPC_JOBNAME=jobname
SA390_JOBNAME=jobname
AFOPER_JOBNAME=jobname
OPSMVS_JOBNAME=jobname
SHUT_OBJECT_SERVER=YES | NO
SHUTDOWN_DELAY=n
SHUTDOWN_CONFIRM=YES | NO
RMF_SERVER_IP=ip address
TCPIP_TASKNAME=taskname
RMF_SERVER_PORT=8802 | port
RMF_SERVER_TIMEOUT=5 | ??
RMF_SERVER_INTERVAL=100 | ??
WTOR_ALERTS=NO | YES
WTOR_DELAY=30 | nnn
WTOR_CLEANUP=900 | nnn
RODM_NAME=name
RODM_USERID=userid
RODM_NOTIFY_METHOD=EKGAOPNF | Method name
RODM_NOTIFY_QNAME=AOPNOTFQ | Notify queue name
PPI=NO | YES
PPI_BUFFER_SIZE=512 | nnn
PPI_BUFFER_QUEUE_LIMIT=1000 | nnn
```

```
PPI_TRACE=NO | YES
SBCS_CODEPAGE=0037|nnnn
DBCS_CODEPAGE=nnnn
LANGUAGE_ID=ENU|xxx
```

Where:

SUBSYSTEM_NAME=TM.. | name | NO specifies the four character name of the MVS subsystem that AOP is to use. If NO is coded, then no subsystem is used and you can only communicate with the TBSM Source/390 Object Pump via MVS modify commands. The .. chars in the default name are replaced by the ID characters from the ACC1Idxx DD card or default to 01 (resulting in a name of TM01) if no ACC1Idxx DD card is coded in the TBSM Source/390 Object Pump JCL.

CONSOLE=TM390.. | name specifies the name of the MCS console that the TBSM Source/390 Object Pump is to use to capture console messages and issue operator commands. The .. chars in the default name will be replaced by the two character suffix (default to 01) from the ACC1Idxx DD card.

CONSOLE_MSCOPE=LOCAL | ALL specifies the scope of the TBSM Source/390 Console. LOCAL specifies that only messages from the MVS image that the TBSM Source/390 Object Pump is running on are captured. ALL specifies that message from all images in the SYSPLEX will be captured.

MAX_TRAPS=2016 | number specifies the number of trap slots that TBSM Source/390 is to reserve for message traps. The number required will vary depending upon the number of traps created and the object being monitored.

TERMINAL_PREFIX=TM390 | prefix specifies the fixed part of the terminal names used as a pool of terminals for 3270 communications. (See the section titled 3270 Pool Terminals for more information.)

NUMBER_OF_TERMINALS=0 | n specifies the number of terminals in the terminal pool. If zero is coded, 3270 communications are not available. Together with TERMINAL_PREFIX, this operand defines terminal names to be used for 3270 communications with Omegamon sessions. (See the section titled 3270 Pool Terminals for more information.)

MODEL2_LOGMODE=D4A32782 | name specifies the logmode name to be used when simulating VTAM sessions using 3270 model 2 (24 x 80) terminals.

MODEL3_LOGMODE=D4A32783 | name specifies the logmode name to be used when simulating VTAM sessions using 3270 model 3 (32 x 80) terminals.

MODEL4_LOGMODE=D4A32784 | name specifies the logmode name to be used when simulating VTAM sessions using 3270 model 4 (43 x 80) terminals.

MODEL5_LOGMODE=D4A32785 | name specifies the logmode name to be used when simulating VTAM sessions using 3270 model 5 (27 x 132) terminals.

LOG_SCREEN=NO | YES specifies whether screen images should be written to the AOPLOG DD. This option can be turned on and off dynamically using the MVS Modify pump,LOGSCREENS option command.

EDI_BUFFER_SIZE=2048 | size specifies the size of the External Data Interface buffer within the TBSM Source/390 Object Pump address space in kilobytes. The size of the buffer required depends upon the amount of data being received by EDI and how quickly EDI can process the received messages within the TBSM Source/390 Object Pump address space.

EXEC_REFRESH=YES | NO specifies whether or not changed REXX EXECs should be refreshed automatically. If YES is coded, REXX checks each EXEC each time it is run to see if it has changed on disk. If NO is coded, REXX does not check for changed EXECs. In addition, the TBSM Source/390 Object Pump will hold REXX EXECs in storage, even when the use count is zero, to improve performance. Changed EXECs can be reloaded using the MVS Modify pump, REFRESH execname command.

SUBSYSTEM_MAXTHREADS=2 | threads specifies the maximum number of concurrent operator commands that the TBSM Source/390 Object Pump can process.

MAXTHREADS_PROMPT=YES | NO specifies whether the TBSM Source/390 Object Pump should issue a message to the operator (which requires a reply), indicating that the maximum number of subsystem threads has been used and giving the operator the opportunity to increase the current number of threads.

OMEGAMON_REFRESH_PERIOD=00:04:00 | hh:mm:ss specifies how often the TBSM Source/390 Object Pump should poll Omegamon sessions for exception information. All fields must be specified and leading zeros must be coded.

OMEGAMON_EXCEPTION_CNTRL=ALL | DELTA specifies whether the TBSM Source/390 Object Pump should send an exception message every time an exception is detected (ALL), or only when new exceptions are detected (DELTA).

TRAP_REPORT=SYSOUT(A) | sysout(class) | dsname specifies where trap reports should be sent.

LOG_FILE=SYSOUT(A) | sysout(class) | dsname specifies where messages logged by TBSM Source/390 Object Pump should be sent. These are in addition to the AOPLOG messages.

LOG_FILE_RECORD_LENGTH=200 | length specifies the LRECL for the log file.

LOG_SPIN_INTERVAL=04:00:00 | hh:mm:ss specifies how often the TBSM Source/390 Object Pump should release the log file for printing.

TRACE=NO | YES YES specifies that the debug messages are to be written to the console. The default is NO. This is an optional operand.

COMMAND_PREFIX=subsystem_name | prefix specifies a command prefix that TBSM Source/390 Object Pump uses to detect commands intended for the TBSM Source/390 Object Pump. This is in addition to the normal Modify interface. If COMMAND_PREFIX is not coded, or allowed to default, then the subsystem name becomes the command prefix for this copy of TBSM Source/390.

OPC_JOBNAME=jobname specified only if OPC EDI interface is required. Specify the name of the OPC Controller and/or Tracker started task, separated by commas if multiple names are utilized

SA390_JOBNAME=jobname specified only if SA390 EDI interface is required. Specify the name of the SA/390 started task, separated by commas if multiple names are utilized.

OPSMVS_JOBNAME=jobname specified only if OPS/MVS EDI interface is required. Specify the name of the OPS/MVS started task, separated by commas if multiple names are utilized.

AFOPER_JOBNAME=jobname specified only if AFOPER EDI interface is required. Specify the name of the AFOPER started task, separated by commas if multiple names are utilized
SHUT_OBJECT_SERVER=YES | NO determines if the TBSM Object Server is to be shut down as a result of the pump being stopped.

SHUTDOWN_DELAY=n the number of seconds the TBSM Source/390 Object Pump should wait before continuing shutdown after a stop command is issued. The default is 0.

SHUTDOWN_CONFIRM=YES | NO specifies whether or not user must confirm before shutting down. The default is YES.

WTOR_ALERTS=NO | YES specifies whether or not WTOR (Write to Operator with Reply) alerts are required. The default is NO.

WTOR_DELAY=30 | nnn specifies the number of seconds that a WTOR message must be outstanding before an alert is raised. The default is 30 seconds.

WTOR_CLEANUP=900 | nnn specifies the number of seconds before a cleanup action is taken to delete old variables from the WTOR shared variable pool. The default is 900 seconds (15 minutes). Reduce this if an excessive number of variables are causing the shared variable pool to become full.

RODM_NAME=name specifies the name assigned to the RODM address space that the TBSM Source/390 Object Pump is to connect to for network status information. If this parameter is not coded, then NO network status data is collected.

RODM_USERID=userid specifies the userid used to sign on to the RODM selected by the RODM_NAME parameter. If this parameter is omitted, the TBSM Source/390 Object Pump job name is used as the RODM user name. If SAF is providing security to RODM, you can code RODM_USERID=' ' to force a blank userid to be used.

RODM_NOTIFY_METHOD=EKGAOPNF | Method name specifies the name of the RODM method that will be used to send status change data from RODM objects to the TBSM Source/390 Object Pump. You may need to change this name if the default name conflicts with an existing RODM method in your installation. In such case, you must rename the supplied RODM method to match the new name that you specify.

RODM_NOTIFY_QNAME=AOPNOTFQ | Notify queue name specifies the name of the RODM notify queue that will be used to transfer data from the RODM address space to the TBSM Source/390 Object Pump. You may need to specify this parameter if the default name conflicts with a name you are using in your installation.

RMF_SERVER_IP=ip address specifies the IP address of the DDS server address space. This is the IP address of the system on which the DDS server resides. If this value is not supplied, the default of the local MVS systems IP address is used.

TCPIP_TASKNAME=taskname specifies the taskname of the TCPIP address space to be used. If this is not supplied, the first active TCP/IP address space is used.

RMF_SERVER_PORT=8802 | port specifies the port number on which RMF responds, This corresponds to the DM_PORT parameter in GPMSRV00 member of the IEFPARM library used by the GPMSERVE proc.

RMF_SERVER_TIMEOUT=5 | seconds specifies the value, in seconds, that TBSM Source/390 Object Pump waits before assuming that the DDS server is unavailable. If a request times out, future collections will not be attempted until at least one RMF_SERVER_INTERVAL has passed

RMF_SERVER_INTERVAL=100 | mintime corresponds to the MINTIME option for RMF Monitor III and should be set to the same value as that set for MF Monitor III.

PPI=NO | YES specifies whether or not the NetView PI receiver should be started. The default is NO.

PPI_BUFFER_SIZE=512 | nnn specifies the size in bytes of the receive buffer for the PPI receiver. The default is 512 bytes. The valid range is 256 to 32768 bytes.

PPI_BUFFER_QUEUE_LIMIT=1000 | nnn specifies the maximum number of records that the PPI buffer can hold. The default is 1000. The valid range is 100 to 10000.

PPI_TRACE=NO | YES specifies whether the records received by the PPI receiver should be written to the AOPLOG file. The default is NO.

SBCS_CODEPAGE=0037|single byte character set codepage ID specifies the single byte character set code page in use on the OS/390 system. Valid values are as follows:

Language	IBM SBCS Code Page	
	Without Euro	With Euro
US English (Default)	0037	1140
Italian	0280	1144
Spanish	0284	1145
French	0297	1147
German	0273	1141
Brazilian Portuguese	0275 0037	1140 1140
Simplified Chinese	0836	
Korea	0833	
Japan (Katakana)	0290	

DBCS_CODEPAGE=double byte character set codepage specifies the double byte character set codepage in use on the OS/390 system. Valid values are as follows:

Language	IBM DBCS Code Page
Simplified Chinese	0837
Korea	0834
Japan (Default)	0300

LANGUAGE_ID=ENU language id specifies the language being used on the OS/390 system. Valid Values are:

Language	Language ID
US English (Default)	ENU
Italian	ITA
Spanish	ESP
French	FRA
German	DEU
Brazilian Portuguese	PTB
Simplified Chinese	CHS
Korea	KOR
Japan (Katakana)	JPN

Installing the CICS Transient Data Exit Program

The CICS Transient Data Exit Program is the means for TBSM to collect CICS TDQ messages. The External Data Interface (EDI) within the TBSM Source/390 Object Pump accepts message data from CICS via the exit.

To install the CICS Transient Data Exit program, perform the following steps:

1. Program AOPCICSx must be assembled and installed into CICS. The following sample jobs may be found in the SGTMSAMP library:

AOPCICS2 - CICS version 2
AOPCICS3 - CICS version 3 and above

2. The program must then be activated as an exit point within CICS. Add the program to the startup PLT AFTER the DFHDELIM entry. The Program List Table (PLT) is a list of programs that CICS runs during startup and shutdown. By putting the AOPCICS3 program into the startup PLT, CICS runs the program during startup, which activates the exit point in CICS, so that CICS transient Data messages are captured.

As well as putting the exit in the PLT, you may also choose to define a transaction code for the program (any 4 character name can be used), logon to CICS and use the transaction to stop or start the exit. This is an optional step.

Once the program is successfully installed, CICS produces the following message at startup: **AOP CICS DATA COLLECTION STARTED**

3. You must code EDI=YES in the TBSM Source/390 Object Pump Start parameters. You may need to increase the EDI_BUFFER_SIZE parameter if you are losing data because of the number of CICS regions sending data to the TBSM Source/390 Object Pump.
4. The CICS regions must be registered to the pump.
5. If you are using the ACC1IDxx card as described in the Running Multiple Copies of Source/390 section, you need to code this in the CICS JCLs as well.

Installing the Operations Planning and Control (OPC) EDI

There are two points of integration with Tivoli OPC. The first point provides a source for capturing updates made to the current plan, made outside the daily planning process. These updates can then be applied to the baseline snapshot created during the initial discovery process (through processing of the Daily Plan report). This reduces the need for continually running reports to gather the changes to the current plan. This is implemented through OPC Exit 7, modified to call TBSM EDI.

The second point is to capture and process alert conditions recognized by Tivoli OPC and forward the information to Tivoli Business Systems Manager. Included in this would be the meaningful 'started' and 'ended' messages that are Tivoli OPC-specific (replacements for the generic IEF403I and IEF404I operating system messages) and include Tivoli OPC-specific information, and to capture exception situations via WTO traps. This point is implemented via both the modified OPC Exit 7, and console monitoring of OPC WTOs.

To implement this process, perform the following steps:

1. Ensure that the ALERTS initialization statement in the Tivoli OPC parmlib has WTO specified as the alert action for all conditions. An example follows:

```
ALERTS   WTO(DURATION
          ERROROPER
          LATEOPER
          RESCONT
          OPCERROR
          QLIMEXCEED)
```

2. Install the TBSM version of Tivoli OPC Exit 7. Possible scenarios for this are:

- a. If you are not currently utilizing OPC Exit 7:
 - Copy program EQQUX007 from the TBSM SGTMMODS dataset into an APF authorized library in the LNKLIST concatenation or as defined by the STEPLIB DD statement in the Tivoli OPC JCL procedure.
 - Update Tivoli OPC EXITS initialization statement in parmlib to call the exit.
EXITS CALL07(YES)
- b. You are currently using the OPC feature of System Automation (SA/390) and are using the SA/390 OPC feature supplied version of Exit 7. That would include the exit driver module, EQQUX007, and two other modules UX007001 and UX007002. As shipped, the SA/390 version of EQQUX007 calls up to 10 programs at that exit point, as if OPC had called them directly. See AOC/MVS V1R4 OPC Automation Programmer's Reference, SC23-3820 for more information on Exit 7 integration:
 - Rename module EQQUX007 in the TBSM SGTMMODS dataset to UX00700n (n = next sequential suffix for module name). Using the above example of two existing modules, UX007001 and UX007002, you would rename the EQQUX007 module in SGTMMODS to UX007003.
 - Copy renamed module UX007003 from SGTMMODS dataset to an APF authorized library in the LNKLIST concatenation or as defined by the STEPLIB DD statement in the Tivoli OPC JCL procedure.
- c. You are currently utilizing OPC Exit 7 for function other than SA/390 – OPC interface.
 - Contact TBSM support for a copy of the EQQUX007 exit driver module to allow for multiple exit 7 modules if this exit is not already available to you via SA/390.

3. Ensure that the following statements reside in the TBSM Source/390 object pump startup parameters: **OPC_JOBNAME=stcname,stcname**
(Where stcname is the started task name for the OPC controller and/or tracker on the system where the pump will run.)

```
EDI=YES
EDI_BUFFER_SIZE=2048 | size
```

(Increase the EDI buffer size based on your use of EDI.)

4. If you are using the ACC1Idx card as described in the section titled, **Running Multiple Copies of Source/390 on an MVS Image**, add this card to your OPC Controller started task.

External Data Interface and Automation Products

TBSM External Data Interface (EDI) for automation products uses cross-memory services to pass messages or exceptions that reside on the same MVS operating systems to TBSM. EDI inserts the time-stamp and sequence number on the message or exception, adjusts the final length, and sends the data to TBSM through Windows NT.

Note: Messages and exceptions are passed directly to the TBSM Source/390 Object Pump by EDI and are not displayed on the system console.

Data Sources

You may employ a diverse number of automation products and applications within an enterprise. Each product or application generates messages and exceptions that are of value and interest for the management of the enterprise and TBSM.

The passing of these messages and exceptions to TBSM through the use of the EDI supports the following automation products and applications.

System Automation for OS/390

SA OS/390 provides monitoring of all enterprise assets as well as timely information about the operational status of those assets. TBSM is fully enabled to intercept the message and exception information generated by SA OS/390.

OPS/MVS

This Computer Associates product facilitates monitoring of the enterprise and provides information about the status or condition of enterprise components. TBSM makes full use of the messages and exceptions generated through OPS/MVS, graphically representing their impact on the respective registered object.

AF/OPERATOR

This enterprise systems management product from Candle Corporation provides current information as to the status of monitored components of the enterprise. The status and information, in the form of messages and exceptions, are processed by TBSM and their impact is recorded on the component object within the TBSM representation of the enterprise.

Main View / Auto Operator

Auto Operator is an application that runs under BMC's MainView product. Auto Operator is used to monitor status and condition of enterprise components. TBSM makes full use of messages and exceptions generated by Auto Operator, and graphically presents their impact on the enterprise.

Messages versus Exceptions versus State Change

Within TBSM messages and exceptions behave differently.

For messages, the combination of the Alert State and Priority attributes dictates the appropriate immediate response from TBSM. Messages are designed to reflect a state change (such as Active, Inactive or Abended). This response is initiated without thresholds having to be exceeded.

For example, a CICS Region monitored by TBSM generates an IEF450I message from that region providing notification that the region has abended. Immediately, TBSM takes into consideration the established Priority attribute value for this message (High) and overlays the representation of the CICS Region with the appropriate Alert State (Red).

With exceptions, thresholds are established for each TBSM Object Type that is subject to exceptions. Thresholds dictate how many exception events need to be exceeded before an alert is overlaid on the exception object. TBSM uses the established Alert State attribute value to determine the type or color (Red or Yellow) of alert to place on the object. In addition, one occurrence of the Alert State and Priority is forwarded and posted to the respective parent object.

For example, the same CICS Region monitored by TBSM generates an MAXT exception that indicates the CICS Region is at or over the allowed number of max tasks. The Alert State attribute for this exception is Yellow and the Priority level is Medium. The Alert State/Priority counter for Yellow/Medium relative to the CICS Region Object Type is incremented by one (1). The resulting number is compared against the established threshold for this exception and the Yellow/Medium combination. Only if the Alert State/Priority counter value is greater than the established threshold is the object overlaid with the Yellow alert.

State changes can be passed for any objects (commonly used for started tasks). By passing state changes, TBSM has the latest status for resources within the enterprise. Also, TBSM can have the correct status for objects after an IPL, during the system SHUTDOWN process, and for scheduled outages.

When passing data (such as state change) to TBSM, objects are auto discovered if they are not on the tree. For example, if started task DFHSM is not defined to TBSM and a state change of BROKEN is sent, TBSM will auto discover it and put it on the tree with a RED icon over the object (DFHSM).

Functional Components

The following functional components facilitate the ability to pass data through EDI to Tivoli Business Systems Manager.

- Application Capture
- REXX Program

Application Capture

A minor modification to the component within the originating external product or application enables the capture and formatting of message and exception information that is generated by the parent product or applications. Once captured, this information passes to the REXX program for further action.

REXX Programs

These programs (GTMEDISA – for SA/390, GTMEDI AF – for AF/OPER, GTMEDIOP – for OPS/MVS, and GTMEDI AO – for Main View Auto Operator) reside within the TBSM installation libraries. These programs format the passed message or exception, providing additional parameters and information to the message/exception as required for use in EDI to TBSM interactions. Upon completion, the REXX programs invoke EDI for further action. GTMEDISA, GTMEDI AF, and GTMEDI AO are sent as compiled programs. GTMEDIOP is sent as source code because of the OPSMVS environment.

An additional REXX member, GTMEDITB, contains all Format, Action, and Data types supported by TBSM. This member is sent as compiled and as source code.

All source code resides in hlq.SGTMSAMP and compiled REXX programs reside in hlq.SGTMEEXEC.

Note: Prior to Tivoli BSM 1.5, the REXX programs for implementing EDI in Automation Tools were AOPEDISA for SA/390, AOPEDIOP for OPS/MVS and AOPEDI AF for AF/OPER. These REXX programs are still supported and supplied in the same libraries as indicated above. However, the new REXX programs are more robust in functionality and it is recommended that you consider converting to them.

All further documentation in this manual regarding the EDI for automation tools, unless specifically stated, only applies to the new GTMEDIXx based code. See Appendix F, titled **Variables Used in Automation EDI REXX Programs AOPEDIxx**, for a list of the variables used in the previous program calls.

EDI

EDI and related functionality are provided with TBSM. The program GTMRXEDI resides in the load module library and adds date and time information as well as assignment of a sequence number. GTMRXEDI also provides the capability to pass the formatted message or exception information to the TBSM Source/390 Object Pump.

Error Handling

When the REXX EDI is not able to send data to the EDI (EDI gets a return code of greater than 0), the following message is sent to the system console:

```
GTMT??101  EDI FAILURE ON sysname. RC=?? AC=?????????
```

where:

sysname

System (OS) where EDI failure occurred

RC Return code from the EDI

AB Abend code from the EDI

?? SA for SA/390, OP for OPS/MVS, AF for AF/OPER, and AO for Main View Auto Operator

When the EDI failure has been resolved, the following message is sent to the system console:

```
GTM??102  EDI PROBLEM RESOLVED ON sysname
```

where:

sysname

System (OS) where EDI problem has been resolved

?? SA for SA/390, OP for OPS/MVS, AF for AF/OPER, and AO for Main View Auto Operator

Automation (Starting/Ending)

The Automation (Starting/Ending) feature is available for the Java-based console, not for the traditional console.

The automation process is very similar to the process of sending messages. The automation routine passes similar information as the REXX routine in the message process.

When automation begins, the EDI receives a message stating “**Automation starting for object xxxxxxxx**”. When automation ends, a message is sent to the EDI stating “**Automation ending for object xxxxxxxx**”.

Note: The grouping (xxxxxxx) indicates a resource name, for example, DFHSM or HSC.

The following example notifies TBSM that automation is in progress for a specific object:

```
'GTMEDISA' START_OF_DATA
Token = "GTM" || rc
'GTMEDISA' TOKEN 07 02/* SA390 MESSAGE EVENT */
'GTMEDISA' TOKEN 05 IP01/* SMF ID */
'GTMEDISA' TOKEN 03 BATCH001/* OBJECT NAME */
'GTMEDISA' TOKEN 08 GTMAUON/* MESSAGE ID indicating automation on */
'GTMEDISA' TOKEN 49 "Automation starting for object BATCH001" /* MESSAGE TEXT */
'GTMEDISA' TOKEN END_OF_DATA TRACEOFF
```

The following example notifies TBSM that the automation procedure has completed for a specific object:

```
'GTMEDISA' START_OF_DATA
Token = "GTM" || rc
'GTMEDISA' TOKEN 07 02/* SA390 MESSAGE EVENT */
'GTMEDISA' TOKEN 05 IP01/* SMF ID */
'GTMEDISA' TOKEN 03 BATCH001/* OBJECT NAME */
'GTMEDISA' TOKEN 08 GTMAUOF/* MESSAGE ID indicating automation off */
'GTMEDISA' TOKEN 49 "Automation ending for object BATCH001" /* MESSAGE TEXT */
*/'GTMEDISA' TOKEN END_OF_DATA TRACEOFF
```

The following example notifies TBSM that the automation procedure has terminated for a specific object:

```
'GTMEDISA' START_OF_DATA
Token = "GTM" || rc
'GTMEDISA' TOKEN 07 02/* SA390 MESSAGE EVENT */
'GTMEDISA' TOKEN 05 IP01/* SMF ID */
'GTMEDISA' TOKEN 03 BATCH001/* OBJECT NAME */
'GTMEDISA' TOKEN 08 GTMAUAB/* MESSAGE ID indicating automation off */
'GTMEDISA' TOKEN 49 "Automation terminating for object BATCH001" /* MESSAGE TEXT */
'GTMEDISA' TOKEN END_OF_DATA TRACEOFF
```


Valid Format Types, Action Types, and Data Types for the EDI

Following are valid Format Types, Action Types, and Data Types for the EDI:

Format Types

- 7 (SA390)
- 8 (OPS/MVS)
- 9 (AFOPER)
- 12 (MAINVIEW AUTO/OPER)

Action Types

- 1 (STATECHANGE)
- 2 (MESSAGE)
- 3 (EXCEPTION)
- 4 (OPC MESSAGE)
- 5 (OPC EXCEPTION)

Data Types

- 5 = SMFID
- 6 = Object_Type
- 3 = Object_Name
- 8 = state_msgid_excpid
- 9 = msg_state
- 10 = application_id
- 11 = input_arrival_time
- 12 = work_station
- 49 = msg_text

TBSM Source/390 Object Pump

Once the message or exception information has been sent to the TBSM Source/390 Object Pump (Object Pump) through EDI, the Object Pump traps the messages or exception, and provides the respective information to TBSM for appropriate action.

Required Application Capture Modifications

To support passing of data from a source product or application to TBSM through the External Data Interface, the following modifications must be made to the application capture within the originating external product or application.

Data must be passed in the following manner: **START_OF_DATA** is passed first, then **FORMAT TYPE** and **ACTION TYPE** are passed. Next, **DATA TYPE** and **DATA** values are passed. Once all **DATA TYPE** and **DATA** values are passed, **END_OF_DATA** is passed. Following is an example of passing a STATE CHANGE via SA/390's REXX EDI:

```
'GTMEDISA' START_OF_DATA
Token = "GTM" || rc
'GTMEDISA' TOKEN 07 01          /* SA390 STATE CHANGE EVENT */
```

```
'GTMEDISA' TOKEN 05 IPO1          /* SMF ID */
'GTMEDISA' TOKEN 06 SUBSYSTEM      /* RESOURCE TYPE */
'GTMEDISA' TOKEN 03 STCTEST1       /* OBJECT NAME */
'GTMEDISA' TOKEN 08 ABENDING       /* STATE CHANGE */
'GTMEDISA' TOKEN END_OF_DATA TRACEOFF
```

Note: TOKEN is used to distinguish between concurrent calls to the EDI. With the use of TOKEN, data is not lost between calls to the EDI. If TOKEN is not used, data may be lost during concurrent calls to the EDI. Every EDI example from this point on makes use of the TOKEN. Following is an example of making an EDI call without the use of TOKEN:

```
'GTMEDISA' START_OF_DATA
'GTMEDISA' 07 01          /* SA390 STATE CHANGE EVENT */
'GTMEDISA' 05 IPO1        /* SMF ID */
'GTMEDISA' 06 SUBSYSTEM   /* RESOURCE TYPE */
'GTMEDISA' 03 STCTEST1    /* OBJECT NAME */
'GTMEDISA' 08 ABENDING    /* STATE CHANGE */
'GTMEDISA' END_OF_DATA TRACEOFF
```

1. The first parameter passed is **START_OF_DATA**. This initializes the REXX EDI to start accepting information.
2. The second parameter passed contains the Format Type and Action Type. In this case, the Format Type is 07 and the Action Type is 01. You can pass Format Type and or Action Type in numeric or character form. You could have passed the second parameter as **'GTMEDISA' 07 STATE_CHANGE** or **'GTMEDISA' SA390 STATE_CHANGE** or **'GTMEDISA' SA390 01**.
3. All parameters passed after the one above contain Data Type and Data. In the example, the Data Type is **05** (or SMFID) and the Data is **IPO1** (name of MVS image). Next Data Type and Data passed are **06** (or **RESOURCE_TYPE**) and **SUBSYSTEM**. After that is **03** (or **OBJECT_NAME**) and **STCTEST1**. Last Data Type and Data passed is **08** (or **EVENT_TYPE**) and **ABENDING**.
4. **END_OF_DATA TRACEOFF** are the last parameters passed. **END_OF_DATA** notifies REXX EDI that there are no more records to be passed. **TRACEOFF** tells the EDI not to print out the record being sent to TBSM PUMP. If **TRACEON** were passed, the record being passed to the PUMP would be printed out. This record is useful for diagnosing a problem with the EDI.

Notes:

1. **Format, Action, Data Types** can be passed in their numeric or character formats. A listing of valid Format, Action, and Data Types is located in the Appendices section of this guide. This data is stored in REXX module GTMEDITB. A compiled version of this module is in hlq.SGTMEEXEC and source is in hlq.SGTMSAMP.
2. When passing text (i.e. message text), please make sure it's in quotes. The following example will post a message, **Test message from SA390** with MessageID SAM001I to object BATCH001:

```
'GTMEDISA' START_OF_DATA
Token = "GTM" || rc
'GTMEDISA' TOKEN 07 02          /* SA390 MESSAGE EVENT */
'GTMEDISA' TOKEN 05 IPO1        /* SMF ID */
'GTMEDISA' TOKEN 03 BATCH001    /* OBJECT NAME */
'GTMEDISA' TOKEN 08 SAM001I     /* MESSAGE ID */
'GTMEDISA' TOKEN 49 "Test message from SA390" /* MESSAGE TEXT */
'GTMEDISA' TOKEN END_OF_DATA TRACEOFF
```

REXX Program Call

You must add the command to call the REXX program GTMEDISA, GTMEDIAF, GTMEDIOP, or GTMEDIAO and to pass the parameters (with their associated values).

The syntax of the program call is dependent upon the data source.

■ SYSTEM AUTOMATION OS/390 (pass STATE_CHANGE)

```
'GTMEDISA' START_OF_DATA
Token = "GTM" || rc
'GTMEDISA' TOKEN 07 01 /* SA390 STATE CHANGE EVENT */
'GTMEDISA' TOKEN 05 IP01 /* SMF ID */
'GTMEDISA' TOKEN 06 SUBSYSTEM /* RESOURCE TYPE */
'GTMEDISA' TOKEN 03 TESTSTC1 /* OBJECT NAME */
'GTMEDISA' TOKEN 08 ABENDING /* STATE CHANGE */
'GTMEDISA' TOKEN END_OF_DATA TRACEOFF
```

■ OPS/MVS (pass STATE_CHANGE):

```
ADDRESS "TSO" "OI GTMEDIOP" START_OF_DATA
Token = "GTM" || rc
"OI GTMEDIOP" TOKEN 09 01/* OPSMVS STATE CHANGE */
"OI GTMEDIOP" TOKEN 05 IP01 /* SMF ID */
"OI GTMEDIOP" TOKEN 06 SUBSYSTEM /* RESOURCE TYPE */
"OI GTMEDIOP" TOKEN 03 TESTSTC1 /* OBJECT NAME */
"OI GTMEDIOP" TOKEN 08 ABENDING /* STATE CHANGE */
"OI GTMEDIOP" TOKEN END_OF_DATA TRACEOFF
```

■ AF/OPERATOR (pass STATE_CHANGE):

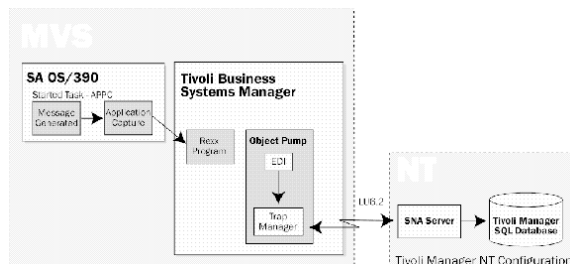
```
CALL "GTMEDIAF" START_OF_DATA
Token = "GTM" || result
CALL "GTMEDIAF" TOKEN 10 01 /* AF/OPERATOR STATE CHANGE */
CALL "GTMEDIAF" TOKEN 05 IP01 /* SMF ID */
CALL "GTMEDIAF" TOKEN 06 SUBSYSTEM /* RESOURCE TYPE */
CALL "GTMEDIAF" TOKEN 03 TESTSTC1 /* OBJECT NAME */
CALL "GTMEDIAF" TOKEN 08 ABENDING /* STATE CHANGE */
CALL "GTMEDIAF" TOKEN END_OF_DATA TRACEOFF
```

■ MAIN VIEW / AUTO OPERATOR (pass STATE_CHANGE):

```
CALL "GTMEDIAO" START_OF_DATA
Token = "GTM" || rc
"GTMEDIAO" TOKEN 12 01 /* MAINVIEW STATE CHANGE EVENT*/
"GTMEDIAO" TOKEN 05 IP01 /* SMF ID */
"GTMEDIAO" TOKEN 06 SUBSYSTEM /* RESOURCE TYPE */
"GTMEDIAO" TOKEN 03 TESTSTC1 /* OBJECT NAME */
"GTMEDIAO" TOKEN 08 ABENDING /* STATE CHANGE */
"GTMEDIAO" TOKEN END_OF_DATA TRACEOFF
```

Event Flow

The following diagram depicts event flow from OS/390 to Windows NT.



Installing the System Automation for SA/390 EDI

TBSM interfaces directly with SA/390 through the implementation of the EDI, which enables an external application to pass text-based messages directly into the TBSM Source/390 Object Pump on the same MVS system. When a status for a resource managed by SA/390 changes, TBSM notifies you of this status change by modifying the TBSM object representing the resource in question.

The following modifications are required to install the SA/390 EDI Interface to TBSM:

1. Edit SA/390 exit AOFEXSTA to add the following code. If this exit is not already installed, it can be found in the SA/390 SINGSAMP dataset member INGEXSTA. Copy the code to a dataset available in the SA/390 DSICLD DD and rename it AOFEXSTA. If the OPC to SA interface is in use, edit exit EVJEXSTA instead of AOFEXSTA.

After the Parse Arg ... statement, add the following code:

```
'GTMEDISA' START_OF_DATA
Token = "GTM" || rc
'GTMEDISA' TOKEN 07 01      /* SA390 STATE CHANGE EVENT */
'GTMEDISA' TOKEN 05 sysname /* SMF ID */
'GTMEDISA' TOKEN 06 restype /* RESOURCE TYPE */
'GTMEDISA' TOKEN 03 resource /* OBJECT NAME */
'GTMEDISA' TOKEN 08 status  /* STATE CHANGE */
'GTMEDISA' TOKEN END_OF_DATA TRACEOFF
```

2. Once the exit has been modified, save it in a dataset that is in the DSICLD DD of the SA/390 started task, preferably, the first dataset in the concatenation.
3. The hlq.SGTMEEXEC library must also be added to the DSICLD DD SA/390 started task.
4. The hlq.SGTMMODS library must be added to the SA/390 started task STEPLIB concatenation.
5. Ensure that the following statements reside in the TBSM Source/390 Object Pump startup parameters:

```
SA390_JOBNAME=stcname
(Where stcname = started task name for the SA task)
EDI=YES
EDI_BUFFER_SIZE=2048 | size
(Increase the EDI buffer size based on your use of EDI.)
```

6. If you are using the ACC1Idx card as described in the section 'Running Multiple Copies of Source/390 on an MVS Image,' add this card to your SA/390 started task.

To have TBSM track Messages and/or Exceptions passed from client written SA/390 REXX execs, perform the following:

1. Edit SA/390 client specific REXX program and include the necessary REXX calls as documented in the section 'Required Application Capture Modifications' and 'REXX Program Call'
2. Once the exit has been modified, save it in a dataset that is in the DSICLD DD of the SA/390 started task, preferably, the first dataset in the concatenation.
3. The hlq.SGTMEEXEC library must also be added to the SA/390 started task DSICLD DD.
4. The hlq.SGTMMODS library must be added to the SA/390 started task STEPLIB concatenation.

5. Ensure that the following statements reside in the TBSM Source/390 Object Pump startup parameters:

SA390_JOBNAME=stcname
(Where stcname = started task name for the SA task)

EDI=YES
EDI_BUFFER_SIZE=2048 | size
(We recommend a size value of 5000 for the SA interface implementation)

6. If you are using the ACC1Idx card as described in the section, *Running Multiple Copies of Source/390 on an MVS Image*, add this card to your SA/390 started task.

Installing the OPS/MVS EDI

TBSM interfaces directly with OPS/MVS through the implementation of the EDI, which enables an external application to pass text-based messages directly into the TBSM Source/390 Object Pump on the same MVS system.

The following modifications are required to install the OPS/MVS EDI Interface:

1. Edit OPS/MVS client specific REXX program and include the REXX calls as documented in the sections 'Required Application Capture Modifications' and 'REXX Program Call
2. Save the REXX code in a library available to the SYSEXEC DD of the OPS/MVS started tasks (both the server and main tasks), preferably the first dataset in the concatenation.
3. The TBSM REXX program GTMEDIOP must be copied from the hlq.SGTMSAMP library to a library available within the OPS/MVS started tasks (both the server and main tasks) SYSEXEC DD. If applicable, GTMEDIOP must then be updated to the pipe symbol 'I' to the appropriate concatenation symbol for your environment. For example, change the pipe to an exclamation point if you are using the German code page on your OS/390 system.
4. The hlq.SGTMMODS library must be added to the SA/390 started tasks (both the server and main tasks) STEPLIB concatenation.
5. Ensure that the following statements reside in the TBSM Source/390 Object Pump startup parameters:

OPSMVS_JOBNAME=stcname,stcname

(Where stcname = started task name for the OPS/MVS Server and main task, the Server task name requires the Server task prefix followed by an * to allow for all task names spawned by the server task. For example, if your Server task name is OSFSRV and your main task name is OPSMVS, code OPSMVS_JOBNAME=OPSMVS,OSF*)

EDI=YES
EDI_BUFFER_SIZE=2048 | size

(A size value of 5000 for the OPS/MVS interface implementation is recommended.)

6. If you are using the ACC1Idx card as described in the section, *Running Multiple Copies of Source/390 on an MVS Image*, add this card to your OPS/MVS Started task.

Installing the AF/OPERATOR and Auto Operator EDI

Make the following modifications to install the AF/OPERATOR and/or Auto Operator EDI Interface:

1. Edit the client specific REXX program and include the REXX calls as documented in the sections 'Required Application Capture Modifications' and 'REXX Program Call.'
2. Save the REXX code in a library available to the SYSEXEC DD of the AF/OPERATOR or Auto Operator started task, preferably the first dataset in the concatenation.
3. Edit the AF/OPERATOR or AOTO OPERATOR started task(s) and add these libraries:
 - hlq.SGTMEEXEC to the SYSEXEC DD concatenation.
 - hlq.SGTMMODS to the STEPLIB DD concatenation.

4. Ensure that the following statements reside in the TBSM Source/390 Object Pump startup parameters:

```
AFOPER_JOBNAME=stcname    for AF/OPERATOR or
AUTOOPER_JOBNAME=stcname  for Auto Operator
EDI=YES
EDI_BUFFER_SIZE=2048 | size
```

(A size value of 5000 is recommended for the Automation EDI interface implementation)

5. If you are using the ACC1Idx card as described in the section titled, Running Multiple Copies of Source/390 on an MVS Image, add this card to your OPS/MVS Started task.

Required NT Changes for EDI

To perform auto discovery of objects using the REXX EDI, use the following sample SQL statements/queries to define objects in the SA390ResourceTypeMap table. (All SQL statements/queries should be performed to the OBJECT database). All REXX EDI interfaces (GTMEDISA, GTMEDIOP, GTMEDIAP, and GTMEDIAO) use this table.

Sample SQL statements/queries:

Select * from SA390ResourceTypeMap (Displays all entries in the SA390ResourceTypeMap table.)

**DefineSA390ResourceTypeMapping @SMFID_PAT='%',
@RESOURCE_TYPE_PAT='%', @OBJECT_NAME_PAT='EDISTC%',
@target_cid='STC', @priority=100** (Defines all objects beginning with EDISTC as started tasks.)

**DefineSA390ResourceTypeMapping @SMFID_PAT='%',
@RESOURCE_TYPE_PAT='%', @OBJECT_NAME_PAT='EDIBAT%',
@target_cid='BATC', @priority=100** (Defines all objects beginning with EDIBAT as batch jobs.)

**DefineSA390ResourceTypeMapping @SMFID_PAT='%',
@RESOURCE_TYPE_PAT='%', @OBJECT_NAME_PAT='EDIDB2%',
@target_cid='DB2', @priority=100** (Defines all objects beginning with EDIDB2 as DB2 subsystems.)

```
DefineSA390ResourceTypeMapping @SMFID_PAT='% ',
@RESOURCE_TYPE_PAT='% ', @OBJECT_NAME_PAT='EDICIC%',
@target_cid='CICS', @priority=100 (Defines all objects beginning with EDICIC as CICS
regions.)
```

Installing the TBSM Omegamon Interface

Following are the requirements for Omegamon (both MVS and DB2) from the mainframe side.

1. The initial Omegamon screen should NOT have any .FGO or .SGO commands on it, as these Omegamon commands may cause Omegamon screens to jump around.
2. Ensure the default Omegamon profile .AS file is defined because TBSM uses this file.
3. The userid used to logon to TBSM must have access to the following Omegamon commands:

EXSY

This is the exception command. If you use the L prefix, this command displays the exception name as well as the text.

.PFK Sets the PF keys for TBSM sessions.

STOP

Logs off the TBSM session.

The CICS file collector process (Omegamon CICS sessions only) also uses the following commands:

TABL Collects CICS table information

FILE Collects CICS dataset information.

4. Ensure the exception settings within the AS profile do NOT have the BOX character set on. (If the BOX character set is on, you will get a line of box characters for the exception.
5. The mode tables for the Omegamon logon are defined in the pump parms, they should exist and they must be SNA log modes. When you log on you are expected to go directly to the Omegamon screen.
6. Enough terminals in the terminal pool for all the Omegamon sessions (one per session). The terminals must be defined correctly in the pump parms:

```
TERMINAL_PREFIX=prefix
NUMBER_OF_TERMINALS=x
```

where the number of terminals 'x' is how many characters must be appended to the prefix to match the applid name defined to VTAM. For example, if the VTAM folks defined applids TM39LU01-TM390LU99, the prefix is TM390LU and the number of terminals is 99. This information must be set correctly to work.

Requirements for Omegamon from the TBSM Workstation

Following are the requirements for Omegamon from the TBSM workstation.

1. If they do not already exist via discovery, you must add each Omegamon as a performance monitor under the OS on which it runs. The performance monitor must be active ('current state' on the Attributes properties page) and registered (status on

Source/390 tab). Ensure you enter values for the applid, vendor, and product on the Source/390 tab for each performance monitor.

2. Update Source/390 tab on DB2 and OS objects with their associated Omegamon applid. DB2 must also be registered and current state must be active.
3. Set values in the user name and password fields at the level necessary. Do this on the Source/390 tab for either the enterprise, complex, OS, or the performance monitor itself. Typically, this is valued at the highest level necessary (by enterprise), but depends on whether you are using the same userid/password for every Omegamon, or different ones depending on where the Omegamons are located.
4. Once this is done, the Connect option should be available on the Source/390 tab on the performance monitor, DB2, and OS. Use the LOGSCREENS command to enable screen logging if you have any problems after this point. Also, you may want to put the sender service in LogLevel 0, so that you can ensure logons (applid, userid, password) being sent up are correct.

TBSM Source/390 Object Server Modify Commands

The Source/390 Object Server supports the following MVS Modify commands:

LOGSWITCH

The LOGSWITCH command requests that the TBSM Source/390 Object Server switch logging of event data from the active log to the idle log, whereby the idle log will become the active log. LOG1 and LOG2 startup parameters reference the datasets that are used for logging the TBSM events.

Syntax: F object_server,LOGSWITCH

READQ

The READQ command requests that the TBSM Source/390 Object Server read the server queue to obtain event messages and send the messages to TBSM.

Syntax: F object_server,LOGSWITCH

TRACEON

The TRACEON command is used to request that debug messages be displayed on the console.

Syntax: F object_server,TRACEON

TRACEOFF

The TRACEOFF command is used to turn off the debug messages. The debug messages are no longer displayed on the console.

Syntax: F object_server,TRACEOFF

SWITCH

The SWITCH command may be used to switch from the primary connection to the backup IP resource. The command may also be used to switch back to the primary connection. The switch command is usable only for the IP interface.

Syntax: F object_server,SWITCH

TBSM Source/390 Object Pump Modify Commands

The Source/390 Object Pump accepts the following MVS Modify commands (F pump,command [options]):

REFRESH

The REFRESH command causes the TBSM Source/390 Object Pump to re-load a REXX EXEC into storage when running with EXEC_REFRESH=NO (no automatic refresh of REXX EXECs). This can be used if a REXX EXEC has changed and needs to be re-re-loaded without restarting the TBSM Source/390 Object Pump. It is not valid if EXEC_REFRESH=YES was coded in the startup parameters, and it has no effect if the EXEC being refreshed was not previously loaded.

Syntax: F pump,REFRESH execname

Where:

Execname is the name of the REXX EXEC to reload.

LOGSCREENS

The LOGSCREENS command allows you to turn 3270 screen logging on or off dynamically while the TBSM Source/390 Object Pump is running. Logged screens are written to the AOPLOG DD.

Syntax: F pump,LOGSCREENS ON | OFF

Where:

ON specifies that screen logging should be started.

OFF specifies that screen logging should be stopped.

If the command is issued without either ON or OFF being specified then the command displays the current status of screen logging.

SHOW

The SHOW command can be used to display the various resources within the source/390 Object pump.

Syntax: F pump,SHOW resource [options]

Where:

Resource I the resource type to display.

Options are any options that are valid for the resource type.

SHOW EXCPS

Displays information about the selected exceptions.

Syntax: F pump,SHOW EXCPS mask [LONG | NOLIST]

Where:

Mask is a generic or specific exception name that is used to select the exceptions name(s) to display. The default is * (all exceptions).

LONG specifies that the long form of the selected exceptions be displayed. The long form includes all the information about the exception and the parameters supplied to the monitoring programs. If the short form is allowed to default then only the exception name, the status and the number of matches is displayed for each exception.

NOLIST specifies that only the summary line is to be displayed which shows a count for the selected exceptions (bask on mask).

SHOW TRAPS

Displays information about the selected traps and/or trap types.

Syntax: F pump,SHOW TRAPS mask type [LONG | NOLIST]

Where:

Mask is a generic or specific trap name that is used to select the trap name(s) to display. The default is * (all traps of the selected type).

Type is the trap type which can be one of WTO,TOD,XOM,XOC,XOI,XO2, CMD or *. The default is * (all trap types).

LONG specifies that the long form of the selected traps be displayed. The long form includes all the information about the trap and the action that it takes. If the short form is allowed to default then only the trap name and the number of matches is displayed for each trap.

NOLIST specifies that only the summary line is to be displayed which shows a count by trap type for the selected traps (bask on mask).

SHOW SESSIONS

The SHOW SESSIONS command displays information about the current 3270 sessions.

Syntax: F pump,SHOW SESSIONS mask

Where:

Mask is a generic or specific session name that is to be used to select the session name(s) to display. The default is * (all sessions).

SHOW POOLS

The SHOW POOLS command displays the names of the currently defined global variable pools

Syntax: F pump,SHOW POOLS [mask]

Where:

Mask is a generic or specific pool name to select the pools to be displayed. The default is * (all pools).

SHOW VARS

The SHOW VARS command displays the variable names and currently assigned values of the selected variables in the selected pool(s).

Syntax: F pump,SHOW VARS varnamemask poolnamemask

Where:

Varnamemask specifies a generic or specific variable name to display. The default is * (all variables in the selected pools).

Polnamemask specifies a generic or specific pool name that is to be used to select the pool(s) to scan for the selected variable names. The default is * (all pools).

Two pools are always defined. They are:

- **System Pool.** Contains Read only variables.

- **User pool.** The user pool always exists, even when empty. Contains work variables used by the source/390 Object Pump.

SHOW MSGCOUNT

Displays the number of console messages received by the console interface.

Syntax: F pump,SHOW MSGCOUNT

SHOW PPICOUNT

The SHOW PPICOUNT command displays the number of application instrumentation (AMI) messages received by the PPI interface.

Syntax: F pump,SHOW PPICOUNT [* | type]

Where:

Type specifies the format type for which the count is to be displayed. The following values are valid types:

- DB2
- IMS
- CPSM
- AMI

SHOW RODM

Displays the current status of the TM/390 pump connection to RODM. Statuses are **connected** or **not connected**.

Syntax: F pump,SHOW RODM

RODM

The RODM command is used to start or stop the Source/390 Object Pump RODM Interface.

Syntax: F pump,RODM START | STOP

Where:

START indicates that the Object Pump should attempt to start communications with RODM

STOP indicates that the Object Pump should stop communications with RODM.

PPI ENABLE

Starts the Source/390 Object Pump PPI Interface.

Syntax: F pump,PPI ENABLE

PPI DISABLE

Stops the Source/390 Object Pump PPI Interface.

Syntax: F pump,PPI DISABLE

PPI STATUS

Displays the status of the Source/390 Object Pump PPI receiver task.

PPI TRACE

The PPI TRACE command will start, stop or display the PPI trace.

Syntax: F pump,PPI TRACE [ON\YES | NO\OFF]

Where:

ON (or YES) specifies that PPI tracing should start. Data received by the PPI listener is written to the Source/390 Object Pump AOPLOG DD.

OFF (or NO) specifies that tracing by the PPI listener should stop.

If an option is not specified, the command displays the current state of PPI tracing.

PPI RESET

Displays or resets the PPI buffer queue limit.

Syntax: F pump,PPI RESET [buffer_queue_limit]

Where:

Buffer_queue_limit specifies the new limit to be applied to the PPI buffer. The valid range is 100 to 10000. If the buffer_queue_limit operand is omitted then the command displays the current setting for the buffer queue limit.

TBSM Source/390 Object Pump Commands

When running the TBSM Source/390 Object Pump with a subsystem, the COMMAND_PREFIX parameter defines a text string that the TBSM Source/390 Object Pump recognizes a command intended for the TBSM Source/390 Object Pump. The string can be any suitable character string and defaults to the subsystem name. For example, if COMMAND_PREFIX=S390 were specified, then any operator command starting with the string 'S390' would be identified as a command string for the TBSM Source/390 Object Pump. In the following text, the string prefix represents the COMMAND_PREFIX character string.

The TBSM Source/390 Object Pump accepts the following commands:

SHOW EXCPS

The SHOW EXCPS command works in the same way as the modify version of the command except that short form output (NOLIST or LONG not specified) is sorted into exception name order.

Syntax: Prefix SHOW EXCPS mask [NOLIST | LONG]

Where:

Mask is a generic or specific exception name that is used to select the exceptions name(s) to display. The default is * (all exceptions).

LONG specifies that the long form of the selected exceptions be displayed. The long form includes all the information about the exception and the parameters supplied to the monitoring programs. If the short form is allowed to default then only the exception name, the status and the number of matches is displayed for each exception.

NOLIST specifies that only the summary line is to be displayed which shows a count for the selected exceptions (bask on mask).

SHOW TRAPS

The SHOW TRAPS command works in the same way as the modify version of the command except that short form output (NOLIST or LONG not specified) is sorted into trap name order.

Syntax: Prefix SHOW TRAPS mask type [NOLIST | LONG]

Where:

Mask is a generic or specific trap name that is used to select the trap name(s) to display. The default is * (all traps of the selected type).

Type is the trap type which can be one of WTO, TOD, XOM, XOC, XOI, XO2, CMD or *. The default is * (all trap types).

LONG specifies that the long form of the selected traps be displayed. The long form includes all the information about the trap and the action that it takes. If the short form is allowed to default, only the trap name and number of matches is displayed.

NOLIST specifies that only the summary line is to be displayed which shows a count by trap type for the selected traps (based on the mask).

SHOW POOLS

The SHOW POOLS command works in the same way as the modify version of the command to display the currently defined global variable pool names.

Syntax: Prefix SHOW POOLS [mask]

Where:

Mask is a generic or specific pool name to select the pool names to be displayed. The default is * (all pools).

SHOW VARS

The SHOW VARS command works in the same way as the modify version of the command to display selected variables from selected global variable pools.

Syntax: Prefix SHOW POOLS varnamemask poolnamemask

Where:

Varnamemask specifies a generic or specific variable name to display. The default is * (all variables in the selected pools).

Poolnamemask specifies a generic or specific pool name that is used to select the pool(s) to scan for the required variable names.

SHOW MSGCOUNT

The SHOW MSGCOUNT command displays the number of messages processed by the console interface task.

Syntax: Prefix SHOW MSGCOUNT

SHOW SESSIONS

The SHOW SESSIONS command displays information about the current VTAM 3270 sessions.

Syntax: Prefix SHOW SESSIONS mask | COUNT

Where:

Mask is a generic or specific session name that is to be used to select the session name(s) to display. The default is * (all sessions)

COUNT specifies that only a count of the currently active sessions is to be displayed.

SHOW COUNTS

The SHOW COUNTS command displays the counts of the total, total good and total bad records received from the Source/390 Object Server address space.

Syntax Prefix SHOW COUNTS

SHOW LOGONS

The SHOW LOGONS command displays the status and result of every VTAM 3270 session logon that has been attempted. Unlike the SHOW SESSIONS command which only shows the status of the current sessions. The SHOW LOGONS command displays logon attempts that have failed or sessions that have ended. Prefix SHOW LOGONS

SHOW RODM

The SHOW RODM command displays the current status of the TM/390 pump connection to RODM. Statuses are **connected** or **not connected**. Cmd_prefix SHOW RODM

Handling the Subsystem Max Threads Condition

Each operator command that is issued is trapped by the TBSM Source/390 Object Pump Subsystem and passed to the TBSM Source/390 Object Pump using an available thread. The maximum number of concurrent commands that can be processed is defined by the TBSM Source/390 Object Pump startup parameter:

SUBSYSTEM_MAXTHREADS=n

If an operator command is trapped and there are no available threads then the TBSM Source/390 Object Pump may, depending upon the setting of the MAXTHREADS_PROMPT startup parameter, issue a Write to Operator with Reply (WTOR) message. The WTOR gives the operator an opportunity to increase the maximum number of threads.

This WTOR is only issued if MAXTHREADS_PROMPT=YES was coded.

If MAXTHREADS_PROMPT=NO was coded then no message is issued and the operator command is missed by the TBSM Source/390 Object Pump. The TBSM Source/390 Object Pump also misses any commands that occur when all the subsystem threads are in use in the future.

The WTOR has the format:

GTM7560I AOP: subsystem_name: MAX THREADS REACHED, REPLY 0-9 TO INCREASE.

If the operator replies 0 (zero), max threads are not increased and if max threads is again reached, this message is not issued again.

In effect, replying zero changes to MAXTHREADS_PROMPT=YES option to MAXTHREADS_PROMPT=NO, for future max threads events.

If the operator replies with a single digit number in the range one to nine then max threads is increased by the specified number. If max threads is again reached, the TBSM Source/390 Object Pump again outputs the GTM7560I message requesting an increase in the max threads. This mechanism prevents the subsystem from depleting ECSA storage in the event of a problem (Each thread takes 168 bytes of ECSA).

Bulk Discovery

The GTMAOPE0 TBSM OS/390 utility and ASIMVSIPLListener TBSM service are used for bulk discovery of many mainframe TBSM source products. Bulk discovery allows you to batch load information from a specific TBSM interface (e.g. OPC, RODM, etc.) into the TBSM Object repository. Refer to the release notes for the source products for specific information about the particular source configuration. In many cases, no changes, other than those described below are needed.

GTMAOPE0 Utility Configuration

The GTMAOPE0 utility transmits any non-VSAM file to Tivoli Business Systems Manager (TBSM)

GTMAOPE0 is a standalone MVS TCP/IP Application that sends data to the TBSM Server that runs the ASIMVSIPLListenerSvc. The following parameters are used with the GTMAOPE0 Utility JCL:

TCPIP_ADDRESS=nnn.nnn.nnn.nnn | TCPIP_NAME=computer_name

Specifies the Internet Protocol (IP) address number or computer name for the connection you use to communicate with the ASIMVSIPLListenerSvc. If the TCPIP_NAME parameter is used, GTMAOPE0 attempts Host Name Resolution to locate the IP address. These are required operands. However, only one of these parameters is required to be specified.

The **nnn** operand of the ADDRESS parameter is to be separated by periods '.' and can be any number in the range from 1 to 255. The computer_name is alphanumeric and the length can range from 1 to 69 characters long. If *traceresolver* is set on in the TCP/IP parameters, TCP/IP debug messages are issued to the job log when host name resolution is attempted. These messages are issued regardless of whether the IP Address is found. The *resolvertimeout* and *resolverudpretries* parameters of TCP/IP control the time specifications for Host Name Resolution processing within the TCP/IP network.

Refer to the TCP/IP manuals for more information regarding the TCP/IP parameters.

Note: If the IP address is setup in the NAT configuration, the NAT IP address must be used. If the NAT IP address is not used, the GTMAOPE0 utility will not connect to TBSM and therefore will not transmit the bulk data.

TCPIP_PORT=nnnnn

Specifies the port number to be used on behalf of the connection. The required operand, **nnnnn** can be in the range from 1 to 32767.

TCPIP_JOBNAME=tcip_jobname

Specifies the TCP/IP address space that is to be used to provide TCPIP services. When not specified, the TCPIP address space is dynamically retrieved. The TCP/IP Jobname cannot exceed 8 characters in length. This is an optional operand.

TIMEOUT=30 | seconds

Specifies the timeout value in seconds in which the TCP/IP services end if there is no response. The default is 30-seconds. The maximum value is 86400 seconds. This is an

optional operand. This parameter works in conjunction with the TCP/IP *resolvertimeout* parameter. For example, when the value of this parameter is less than the *resolvertimeout* parameter, the IP service requests time out prior to TCPIP ending the outstanding service request.

Note: Refer to the TCP/IP manuals for more information regarding the *resolvertimeout* TCP/IP parameter. Depending on the volume of data received, you may need to increase the TIMEOUT value.

CODEPAGE=codepage_number | 037

Specifies the codepage number to the ASIMVSIPLListenerSvc uses to convert the data to the appropriate American National Standard Code for Information Interchange (ASCII) symbols. This is an optional control card. The default is US codepage 037.

TEXT=NO | YES

Specifies the transmission protocol is either binary or text mode. When NO is specified, the data is transmitted as binary data. When YES is specified, the DATA_SEPARATOR byte value is used to separate each logical record. This is an optional control card. The default is NO.

DATA_SEPARATOR=byte_value | x'25'

Specifies the byte value to be inserted between every logical data record when running with **TEXT=YES**. This is an optional operand.

When not specified and **TEXT=YES** is requested, the default byte value is x'25'. A decimal value may be specified to identify this byte value. The range is from 1 to 255.

COMMAND=command_name

Specifies a command name sent to the ASIMVSIPLListenerSvc for triggering the command execution on NT to process the data after all the data has been received. The Command_name is an alphanumeric string that can range from 1 to 72 characters long. This is a required operand.

ASIMVSIPLListener TBSM Server Configuration

The ASIMVSIPLListener service runs on the TBSM SQL server. This service must be configured for each OS that is to send data to the SQL server. The ASIMVSIPLListener service is responsible for receiving data sent from the TBSM GTMAOPE0 program.

The following Windows Registry Options are available:

- This setting defines the port address to communicate with GTMAOPE0. This is specified on the TCPIP_PORT control card of GTMAOPE0. The default value is 1021.

Key	HKEY_LOCAL_MACHINE \ SOFTWARE \ Accessible Software, Inc. \ Access1 \ 1.0 \ Components \ ASIMVSIPLListenerSvc \ Settings
Value Name	Port
Value Type	DWORD
Data Value	0x000003fd (1021)

- This setting defines the command to be triggered on the Windows NT Server by GTMAOPE0. This is specified on the COMMAND control card of GTMAOPE0. Below is an example for CA7 discovery. Normally, the 'Value Name' and 'Data Value' are set by the TBSM installation program. You should never change the 'Data Value' without contact Tivoli Support. If the required values are not already part of your TBSM installation, refer to the release notes for the interface you are trying to configure for more details on this Key.

Key	HKEY_LOCAL_MACHINE \ SOFTWARE \ Accessible Software, Inc. \ Access1 \ 1.0 \ Components \ ASIMVSIPLListenerSvc \ Settings \ CommandAliases
Value Name	CA7DISCOVERY
Value Type	String
Data Value	sh ca7Enqueue.ksh -i%s -oc:/tivolimanager/data/ca7/%DS -r<prefix>

- This setting defines the IP clients that are authorized to run GTMAOPE0. The 'Value Name' should be set to the IP address of the mainframe sending the data to NT. The string does not need to be valued, but may contain the DNS name of the host for readability. The 'Data Value' is a string of zero length.

Key	HKEY_LOCAL_MACHINE \ SOFTWARE \ Accessible Software, Inc. \ Access1 \ 1.0 \ Components \ ASIMVSIPLListenerSvc \ Settings \ ValidClients
Value Name	<host>
Value Type	String
Data Value	

GTMAOPE0 Problem Analysis

The following DDNAMES are used as switches to enable tracing and logging information during the transmission of data to TBSM. When these DDNAMES are not used, tracing and logging information is not generated.

TRACE DD coded in the JCL requests that the data records transmitted to TBSM are to be written to the specified file. Depending on whether the TRACE keyword is used, the output is formatted as described by the TRACE keyword operand in the GTMAOPE0 Utility JCL.

TRACE=BUFFERS | **RECORDS**

Specifies whether the data transmitted to TBSM is written to DDNAME TRACE based on the operand provided. BUFFERS indicate that each populated buffer is written to the trace file. RECORDS indicate that each record is individually written to the trace file. BUFFERS operand is the default.

LOG DD coded in the JCL requests that logging information is written to the file specified on this ddname.

The following message is written to the log file specified on the LOG DD statement:

BUFFER COUNT: buffer_id# **PHYSICAL BUFFER:** xmit_bytes **LOGICAL BUFFER:** logical_buffer_size

-
- BUFFER COUNT is the number representing the additional sequence of each buffer as the buffers are transmitted to TBSM. This value always increments by 1.
 - PHYSICAL BUFFER indicates the number of bytes that are transmitted over the IP connection to TBSM.
 - LOGICAL BUFFER indicates the number of bytes that the TBSM ASIMVSIPLListenerSvc service is to process. This value is always 2 bytes less than the PHYSICAL BUFFER value.

The ASIMVSIPLListenerSvc

The TRACE and LOG DDNAMES are used in conjunction with the ASIMVSIPLListenerSvc running with Log Level 0. Informational messages identifying the handling of the buffers received from the GTMAOPE0 utility are found in the log file used by the TBSM MVSIPListenerSvc.

Tivoli Support may require documentation from the following sources:

- The LOG file output that is generated from the GTMAOPE0 utility
- The TRACE file output that is generated from the GTMAOPE0 utility.
- The LOG file generated by the ASIMVSIPLListenerSvc running on Windows/NT

Installing LU6.2 Support

The APPC transport mechanism for sending data between TBSM Source/390 and Windows NT for TBSM uses LU6.2 services over Microsoft's Host Integration Server.

To use Host Integration Server with Windows NT, the following areas require configuration setup.

- Configure VTAM environment
- Install and Configure Host Integration Server
- Install and Configure Host Integration Client

The administrator should be familiar with MVS/VTAM and Windows NT environments.

Configuring VTAM environment using PU2.1 Node

Tivoli requires a PU2.1 end node defined with an independent LU that can be accessed by all OS/390 systems within the enterprise between Source/390 and TBSM on Windows NT.

Consult your VTAM system programmer to provide a physical configuration using an ILU between VTAM and Microsoft's Host Integration Server. A sample configuration for a switched VTAM connection is below. This example can be used in a Token Ring or Ethernet environment. Be sure to code unique names and IDNUM when configuring the VTAM definitions.

Installation Steps

```

*/ * -----
*/ * SNA SERVER PU FOR TM/390
*/ * -----
*/ *
SWTM3901 VBUILD TYPE=SWNET
*/ *

```

```
TM390PU1    PU  PUTYPE=2,ADDR=C1,MAXDATA=1456,      -  
              IDBLK=05D,IDNUM=00002,CPNAME=TM390CP1,  -  
              DLOGMOD=DYNAMIC,USSTAB=USSS  
*/  
TM390LU1    LU  LOCADDR=0
```

Microsoft's Host Integration Server can support Token Ring, Ethernet, and FDDI connections within a VTAM (SNA) environment. Open System Adapters (OSA) and 3172 Nways controllers may also be used with the listed protocols.

7

Installing Host Integration Server

This chapter describes how to install and configure Microsoft Host Integration Server (HIS) so that it communicates with the Object Server component of Source/390. For information about installing and configuring Microsoft's SNA server, see Appendix H of this book.

If you are installing HIS on Windows NT, refer to the Microsoft Host Integration Server README document for information about installation prerequisites. Required prerequisites are available for download from Microsoft's Web site. If you are installing HIS on Windows 2000 Advanced Server, no additional prerequisites are required.

Host Integration Server Security

HIS client applications access the mainframe Object Server task via the HIS server machine. For this reason, the Microsoft Host Integration Server client requires authority to access the HIS server machine and vice versa. For this to be accomplished, the following important NT security issues must be addressed:

- The Microsoft Host Integration Server and the Host Integration client must be installed with the same userid and password.
- If you install the Microsoft Host Integration Server and the Microsoft Host Integration client with a userid that already exists, you must manually give that userid authority to act as part of the operating system.
- If you logon to the Microsoft Host Integration client machine with a userid and password that are different from what you used to install the Microsoft Host Integration Server software, the userid and password must exist with the same password on the Microsoft Host Integration Server server machine.

Installing the Host Integration Server

To install the Microsoft HIS Server:

1. Run the Microsoft Host Integration 2000 Setup program and select **Install Server**. Depending on the configuration of your operating system, you may get an error message stating that pre-reqs for MQ Series bridge and Security Integration are not in place. Select **Yes** to continue if you get an error for these specific options.
2. Follow the instructions on each screen as you proceed through the installation, unless otherwise noted below, use the default installation options for each screen.
3. On the **Select Features** dialog, choose the following options by clicking the icons and select **X Entire Feature will be unavailable**.
 - Data Integration

- Application Integration
 - Under Application and Tools: **5250 client**, **AFTP Client**, and **Host Accounts**.
4. Refer to the **Host Integration Server Security** topic for information on what to enter on **Services Account** dialog.
 5. When you reach the **SNA Resource Location Wizard**, click next, and enter a **SNASUBDOMAIN** that is different from your NT domain or computer name, such as computernameSNA. If you fail to enter a unique name on this screen, you will have difficulty later when configuring Microsoft HIS. Select **Primary** as the server role. Select the default options for the remaining wizard screens.

You are now ready to configure Microsoft Host Integration Server. After you complete the configuration, you must set the service **SNASERVER** to automatically start on reboot. You can do this through the system services manager

Configuring Host Integration Server 2000

SNA Server Manager is the managing component that sponsors sessions on behalf of users between HIS server and VTAM on the mainframe. If you have questions, answers to configuring SNA Manager can be found in the help facilities provided by HIS Server.

Before getting started, work with your VTAM systems programmer to complete the following chart. Many of these values are required during the configuration of HIS server and client.

Parameter	Value*	Description
VTAM Major Node Names		Applid and PU major node names
Network Name		NETID(D NET,VTAMOPTS,OPTION=NETID)
Control Point Name		CPNAME in PU major node
PU Name		name of PU as defined to VTAM
Local Node ID(XID)		IDNUM/IDBLK from PU major node**
Local/Partner LU alias/LU name		LU with LOCADDR=0 in PU major node
Remote LU alias/LU name		Applid name for TBSM (one for each host)
Remote Network Address		MAC address of network interface card**
Remote SAP address		SAPADDR of mainframe network interface (default is 4)**
SNA Server Machine Name		name of NT machine where SNA Server is installed
* Indicates that these values are entered site specific		
** Indicates parameters for switched connections		

Install Microsoft's HIS Server on the Windows NT workstation that is physically connected with a Token Ring or Ethernet card. This workstation should be locally or remotely attached to MVS.

The following steps identify the tasks involved in configuring the HIS Server sections. HIS Server is designed like Windows Explorer, using a tree-like hierarchy. Each section requires an insertion of an object type that is to be created and configured.

A token ring network configuration is provided here to show a sample of how to configure the HIS Server. (For additional information, refer to the HIS Server help.)

The tree structure of HIS Server that is to be configured is:

Servers

Link Services

SNA Service

Connections

Local APPC LUs

Remote APPC LUs

APPC Modes

CPIC Symbolic Names

Under servers, locate the name of the server in which HIS server is installed and make the following configuration changes. HIS Server is a traditional Windows application and you can access the properties menu by right clicking any object and then selecting the properties button.

1. **Update a SNA Service.** You are required to provide the Network Name and Control Point Name of the VTAM environment to which the SNA manager will be connected. Edit **SNA Service** properties to add the Network Name and Control Point Name.

Title	Name for SNA Service
Comment is optional	
Network Name is required	
Control Point Name is required	

2. **Insert a Link Service** on behalf of the SNA Service created using the protocol that supports the type of physical connection being used. For example, Token Ring or Ethernet can use the DLC 802.2 protocol.

(The protocol used must be installed on the machine. If it is not, you will get an error message indicating that the protocol is not installed on the computer.)

Title	Name for Link Service
Adapter name of Token Ring / Ethernet	
SAP	x04
Check	Fixed SAP
Do not check	Allow Link Service to be distributed

3. **Insert a Connection for the Link Service (right-click SNA service and select New-->802.2).**

■ General Tab

Title	Name for Connection Service
Name	Name for connection (PU name is often a useful name here).
Select service name of link service	
Comment is optional	
Remote End	Host System
Allowed Directions	Both Directions
Activation	On Server Startup

■ Address Tab

Title	Name for Connection Service
Remote Network Address	12 Byte Mac Address
Remote SAP address	Remote SAP address from VTAM

■ System Identification Tab

Title	Name for Connection Service
Local Node Name options:	
Network Name	See #1, Insert a SNA Service
Control Point Name	Cpname defined on the PU2.1 node
Local Node ID	IDNUM and IDBLK defined on the PU2.1 node
XID type	Format 3
Remote Node Name options	Do not fill in.
Compression Type	None

■ 802.2 DLC Tab

The default settings are sufficient.

At this point, if the physical connection is set up and the definitions within SNA Server are correct, the connection should activate upon starting the SNA Service.

There is an 'Active' status displayed on the SNA Service and Connection name when started.

4. Insert a Local APPC LU

■ General Tab

Title	Name for Local APPC LU
LU Alias	The name of the Independent LU
Network Name	same name as provided in #1
LU Name	same name as LU Alias
Comment is optional	

■ Advanced Tab

Title	Name for Local APPC LU
Member of default outgoing local APPC LU Pool	Check
Timeout for starting invocable TPs	600 seconds
Implicit Remote LU	None
LU6.2 Type	Independent
LU6.2 Resync Service	Do not fill-in.

There is only one Local APPC LU defining the PU2.1 node residing in VTAM.

5. Insert a Remote APPC LU

■ General Tab

Title	Name for Remote APPC LU
Connection	Assign connection name to be used
LU Alias	Remote APPC LU
Network Name	same name as provided in #1
LU Name	same name as LU Alias
Uninterpreted Name	same name as LU Alias
Comment is optional	

■ Options Tab

Title	Name for Remote APPC LU
Parallel Sessions	Check
Implicit Incoming Mode	None
Session level security	No
Enable SyncPoint	Do not check.

Setup one Remote APPC LU definition for each APPC applid defined for the OS/390 server address spaces. If there are 8 MVS systems in which TBSM for Windows NT is to collect data from, then there should be 8 Remote APPC LU definitions defined within this section.

6. Insert a APPC Mode – insert the mode(s) name defined in VTAM

■ General Tab

Title	Name for Modetab
Mode Name	Your installation's VTAM mode entry name
Comment is optional	

■ Limits Tab – Must match the VTAM applid definitions

Title	Number
Parallel Session Limit	4
Minimum Contention Winner Limit	1

Title	Number
Partner Min Contention Winner Limit	1
Automatic Activation Limit	0

- **Characteristics Tab** – RU sizes must match the VTAM mode table

Title	Number
Pacing Send count	7
Pacing Receive count	7
Max Send RU size	1024
Max Receive RU size	1024

- **Partners Tab**

Follow the Add instructions to associate the LU pairs, Local and Remote LU names.

- **Compression Tab**

The default settings are sufficient.

7. Insert a CPIC Symbolic Name

There are two symbolic names to be inserted. The name ACC1RCV is associated with the process name for data, which is received by the client on behalf of SNA Manager. The name ACC1RECV is associated with the process name for data sent by the client to OS/390. ACC1RECV is only used for the APPC Upload function. Clients using the FTP upload function do not require ACC1RECV.

ACC1RCV

- **General Tab**

Title	Name for CPIC Symbolic name
Name	ACC1RCV
Comment is optional	
Conversation Security	None
Mode Name	Your installation's VTAM mode entry menu

- **Partner Information Tab**

Title	Name for CPIC Symbolic name
Check Application TP	Enter ACC1RCV in the box to the right
Partner LU Name Alias	Local APPC LU name

ACC1RECV

- **General Tab**

Title	Name for CPIC Symbolic name
Name	ACC1RECV
Comment is optional	
Conversation Security	None

Title	Name for CPIC Symbolic name
Mode Name	Your installation's VTAM mode entry menu

■ Partner Information Tab

Title	Name for CPIC Symbolic name
Check Application TP	Enter ACC1RECV in the box to the right.
Partner LU Name Alias	Remote APPC LU name

SNA Server Manager is now configured and ready to be used by a SNA Manager Client. When making changes to SNA Server, the connection and SNA Service must be stopped. After the changes are made, the configuration must be saved before restarting the SNA service and connection name.

Installing Microsoft Host Integration Client

To install the Microsoft HIS Client:

1. Run the Microsoft Host Integration 2000 Setup program and select **Install Server**. Depending on the configuration of your operating system, you may get an error message stating that pre-reqs for MQ Series bridge and Security Integration are not in place. Select **Yes** to continue if you get an error for these specific options.
2. Follow the instructions on each screen as you proceed through the installation, unless otherwise noted below, use the default installation options for each screen.
3. On the **Select Features** dialog, ensure that the following options are the only ones that are selected for installation:
 - SNA Application support (Note that all options underneath this must **NOT** be installed.)
 - Applications and Tools
 - **3270 Client**.
4. On the **Services Account** dialog, enter the same userid and password used during the HIS Server installation.
5. On the **Locate Servers** dialog, select **Locate servers by name** and **Add** the name of your HIS Server.
6. Select the defaults for the remaining dialogs.

Configuring the Host Integration Server Client

After installation, there are two steps to configure for establishing communication to each MVS system.

Step 1

Install the **TPSTART.exe** program to run on the client when the Windows NT workstation is started.

The TPSTART program is a background task that must be running enabling the SNA client to communicate with the SNA Manager using LU6.2 Services.

The TPSTART program can be found in the SNA\System folder where the SNA Client is installed. Set up a shortcut in the Startup folder for the workstation.

Step 2

Install the OS/390 Components on the Event Handler server.

Before attempting to configure the OS/390 components for your TBSM environment, the operating systems you wish to add must be in the TBSM database. Additionally, to send data from TBSM NT to OS/390 TBSM, the **Source/390** box must be checked on the **Source/390** tab of each operating system.

Event Handler Configuration Options

The two configurations for the TBSM communications services are:

- Configuration 1: All communications services are running on the SNA Client machine, known as the Event Handler. Generally, this configuration is used for installations with fewer than 35 operating systems.
- Configuration 2: The listener and event handler processes are running on the SNA Client machine, and the sender services are running on the SNA server machine. Generally, this configuration is used for installations with more than 35 operating systems.

Configuration 1 is considered the standard installation method. If your installation involves more than 35 operating systems, sender services should be created on the SNA server machine. If your installation involves more than 35 operating systems, follow instructions in the Installing Event Handler (Configuration 2) section.

Notes:

1. TPSTART is only required where the Listener Service is installed.
2. Never install the Listener on more than one client pointing to the same HIS server.
3. You must reboot the Event Handler server the first time you run makemvscomponents or anytime you change the local LUNAME (-t parameter).

Installing Event Handler Server (Configuration 1)

On the Event Handler server, the MakeMVSComponents shell script has been provided to perform the following:

- Create the LU6.2 or IP Sender Service for each OS
- Create the File Receiver Service for each OS
- Create the Event Handler Service for each OS
- Update the listener registry for each OS (LU6.2 or IP).

Run this script on the Event Handler server from the TivoliManager\bin directory. Run the script for each mainframe operating system (OS) in your environment. Before running the script, each mainframe OS must have been created within the database. You may obtain the syntax for the MakeMVSComponents script by issuing **sh makemvscomponents -h** from a command prompt on the Event Handler server.

For Configuration 1, use the default value for the component_listing.

Installing Event Handler Server (Configuration 2)

The difference between Configuration 1 and Configuration 2 is that the components are split between the SNA server and the Event Handler server. In Configuration 2, you use the **-c** option in the **makemvscomponents** script to install the sender service(s) on the SNA server and the remaining services on the Event Handler server. Examples follow.

Note: The **ASIEnqueueProxyServer** must be running on the same server where the **ASIMVSSenderSvc** is running. If the **ASIUploadRuleSvc** is installed here, it must be disabled. There should be only one **ASIMVSUploadRuleSvc** per TBSM system environment.

Example 1. Install only Event Handler and Listener Services for an OS named M2CPUA on M2/LPARA/CPUA on the Event Handler server:

```
sh makemvscomponents -BM2CPUA -O'M2/LPARA/CPUA' -cEH,LS,FR
-dUSASI01.NJ1TM390
```

Example 2. Install only Sender Service for an OS named M2CPUA on M2/LPARA/CPUA on the SNA server:

```
sh makemvscomponents -BM2CPUA -O'M2/LPARA/CPUA' -cSS -xNJ1TM390
```

After you run these scripts, start the new Event Handler and Sender Services that were created for the mainframe operating systems. The Sender Service will not successfully start if the SNA Server configuration for this OS is not in place.

Validating LU6.2 Support

Perform the following steps to validate that a physical connection exists between Tivoli Manager Source/390 and Microsoft's Host Integration Server.

1. Open the SNA Manager Application to view whether the connection is Active. If the connection is not active, start the connection. (Reference Microsoft HIS Server help tutorials)

Once the physical connection is verified, the MVS listener processes are ready to be used with TBSM for Windows NT.

2. Start Tivoli Manager Source/390 on an OS/390 system. The MVS listener program on the HIS client workstation associated with that OS/390 system should start running automatically. (Reference Source/390 Install and configuration)

The following methods can be used to determine if a connection exists between SNA Server and an MVS system:

- Use the APPC Viewer on HIS Server Manager to view the sessions connected. (Reference HIS Server help tutorials)
- Issue D NET,ID=luname,E on the MVS console to view the sessions established from VTAM's perspective. (Reference VTAM Operators Guide)
- Browse the Logs Folder within the TBSM Windows NT directory on the SNA Client workstation searching for files beginning with LS and MVSL_. If these files exist, then the SNA Server Manager invoked the TP program (ACC1RCV) on the client machine. There is a log file generated for each MVS Listener invoked by SNA Server. The status is indicated in a message within the log file.

Installing Event Handler Server Using IP (Configuration 3)

This section includes an example for installing TBSM to use IP communication services. The example installs TBSM NT/2000 services for operating system CPUTA under LPAR LPARA. Use the `-h` option of the **makemvscomponents** script to obtain help information.

```
sh makemvscomponents.ksh -BM2CPUTA -O'M2/LPARA/CPUTA'  
-wIBMMVS.HOSTA.COM:1023 -v1022
```

Where 1023 is the port number that must be specified on the TCPIP_SERVER_PORT card for TBSM object server.

Where 1022 is the listening port number that must be specified on the TCPIP_PORT card for TBSM object server.

Notes:

1. The IP communications interface is installed to provide the codepage tables.
2. If you are migrating from LU6.2 to IP, run **makemvscomponents** with the `-r` option to remove components, then re-run **makemvscomponents** to install the components to use IP.

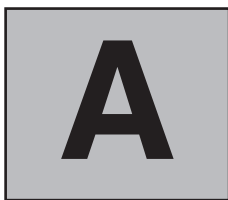
Adding Additional Operating Systems

If a new operating system (OS) is installed into the environment after the TBSM installation is complete, the following may be done to connect the new OS to TBSM.

1. Install the three started tasks (TBSM Source/390 Object Pump, TBSM Source/390 Object Server, and TBSM Source/390 Dataspace) on the new operating system. APF authorize the load library. Copy the startup parameters from an existing functional system. Ensure all MVS tuning requirements are met (e.g. non-swappable and appropriate dispatching priority).
2. Install and customize any necessary TBSM interfaces to other OS/390 products, such as OPC, CA-7, SA/390, CICS, and DB2.
3. Make the following VTAM changes (reference the VTAM configuration section of this book for further information);
 - a. Define the TBSM Source/390 Object Server applid to VTAM on the new system. Ensure that this applid has access to the SNA Server LU (for example, add any CDRSCs needed or update SME exit as required). The applid and the LU should both be able to initiate sessions with one another. Update the Object Server startup parameters to use this applid.
 - b. Define Omegamon applids if you use this performance monitor to capture data. Remember that you need one applid defined for every CICS, DB2, or MVS Omegamon that will log on. Update the Object Server startup parameters to reflect the applid definitions installed.
 - c. Install the logmode table for TBSM. Ensure that the logmode name used is indicated in the Object Server startup parameters, makemvscomponents shell script, and the applid definition.
4. Update HIS Server as follows;
 - a. Insert a Remote APPC LU representing the new Object Server applid.

-
- b. If you are using a different logmode table than the other installed operating systems, insert that table name as an APPC mode and configure it to match the VTAM definitions.
 - c. If you are using an existing logmode, update the Partners tab to reflect the new relationship between local and remote LU.
These changes require a restart of SNABASE on this machine and the Event Handler machine. We recommend that you reboot the SNA server machine and then reboot the Event Handler machine at this time.
5. Update the Database to include the new OS. On the Source/390 tab, make sure Source/390 is enabled. You cannot proceed to Step 5 without adding the OS to the Database. Also, please be sure to place the new OS in the appropriate Line of Business within the LOB called SYSPLEXES.
 6. On the Event Handler PC, use the makemvscomponents script (as documented in the section titled Installing the MVS Components on the Event Handler Server) to create the necessary components for the new OS. Once you have run these scripts, start the new Event Handler and Sender Service that were created for your OS.

You are now ready to test connectivity to the new OS. Refer to the section titled Validating LU6.2 Support for procedures.



Hardware

The successful implementation of Tivoli Business Systems Manager (TBSM) involves the cooperation and interfacing of several components. These components fall into two categories: Hardware and Software. Hardware components provide the physical and logical architecture to support the operations of TBSM. The hardware elements pertain to the mainframe, server-based, and workstation components. Software elements are installed on the hardware to provide operational support, such as databases, communications, and execution for TBSM operations.

This appendix provides detailed information about the hardware and software elements needed to support implementation of TBSM. Information includes a broad definition of each category, as well as specific configurations required for each element.

Differences between Enterprise Edition and Distributed Edition

Hardware and Software requirements for both the Tivoli Business Systems Manager Enterprise Edition and the Distributed Edition are documented in this section. Although most of the hardware and software requirements for both editions are identical, differences are noted. The most prominent difference is in the number of servers recommended, which causes some consolidation of software components. For example, the Enterprise Edition recommends that one server contain Propagation components and one server contain the Application Server; while in the Distributed Edition, both Propagation and the Application Server run on one server.

NT Servers (Enterprise Edition)

Installation of several NT-based servers is critical to the success of TBSM. Each of these installed servers performs specific processing or functionality in support of Tivoli Business Systems Manager.

Six servers provide for the functional operation of TBSM Enterprise Edition at the NT-level. These operations are:

Propagation Server

Facilitates the escalation of alerts from messages and exceptions to Tivoli Business Systems Manager.

SQL Server

Provides for SQL Server execution and contains the main object repository used by Tivoli Business Systems Manager.

History Server

This Server maintains an audit trail of all actions and events acted upon by Tivoli

Business Systems Manager. In addition, the History Server provides the platform of operations for Internet Information Services.

Application Server

Serves as the execution environment for all workstation-based application programs and procedures that are used by TBSM.

Event/File Handler Server

Provides the environment where data is placed in File-Backed Queues for insertion into the database.

SNA Server

Provides a SNA management environment for communications between network components and the mainframe.

The remaining servers are used for:

- Testing
- Quality Assurance

Optimally, these servers should be enabled to support Testing and Quality Assurance (QA) operations for the TBSM implementation.

It is strongly recommended that the acquisition, installation and configuration of these servers be accomplished. While it is possible to achieve a successful TBSM installation without Testing or Quality Assurance servers, increased productivity and a more efficient implementation process offset the additional hardware cost.

NT Servers (Distributed Edition)

The installation of several NT-based servers is critical to the success of TBSM. For each implementation, a total of two servers is recommended. Each one of these installed servers performs a specific group of processing or functionality in support of TBSM.

Two servers provide for the functional operation of TBSM - Distributed Edition at the NT-level. These operations are:

SQL Server

Provides for SQL Server execution and contains the main object repository used by TBSM.

Application Server

This server performs a variety of tasks. The Application Server acts as the execution environment for all workstation-based application programs and procedures, while propagation processes the exceptions and messages that are being input into the system by the Agent Listener process.

NT Workstation

To provide for both administrative and operational support, several NT-based workstations must be obtained. For each implementation, three workstations are recommended. These workstations provide a vehicle for user intervention and interaction with TBSM, as well as facilitate server management operations.



Enterprise Edition Hardware & Software

Following are the hardware and software requirements (production) for the Enterprise Edition of TBSM.

Hardware

The following table displays hardware requirements.

Propagation and Application Server	SNA Server	History Database Server	SQL Database Server	Event Handler Server for OS/390
Single Pentium III, 866MHz, Dual Capable	Single Pentium III, 866MHz	Dual Pentium III Xeon, 866MHz, 1 MB cache, Quad processor capable	Dual Pentium III Xeon, 866MHz, 1 MB cache, Quad processor capable	Single Pentium III, 866MHz, Dual Capable
1GB RAM	1GB RAM	4GB RAM	4GB RAM	1GB RAM
System Drive: RAID1, 18GB	System Drive: RAID1, 18GB	System Drive: RAID1, 18GB Partition	System Drive: RAID1, 18GB	System Drive: RAID1, 18GB
Data Drive: RAID1, 18GB	Data Drive: RAID1, 18GB	Data Drive: RAID5, 108GB Partition	Data Drive: RAID5, 72GB	Data Drive: RAID1, 18GB

Note: If you're running NT 4.0 with a 4GB system drive, TBSM and its prerequisite software should be installed on an alternative drive with 18GB of free space.

All servers must have:

- Redundant Network Interface Card (NIC)
- Redundant Power Supply

Software

All servers must have:

- Windows 2000 Server and Service Pack 1 **or** NT Server 4.0 and Service Pack 6a
- MKS Toolkit 7.0
- NT 4.0 Resource Kit Supplement 3

Note: This resource kit is also required on Windows 2000. Do not use Windows 2000 Resource Kit.

-
- Internet Explorer (IE) Version 5.0

SQL Database Server must have:

- NT 4.0 Enterprise Edition, if using NT 4.0
- SQL 7.0 Enterprise Edition and Service Pack 2
- Microsoft Host Integration Server / SNA Server 4.0 and Service Pack 2 Client

History Database Server must have:

- NT 4.0 Enterprise Edition, if using NT 4.0
- SQL 7.0 Enterprise Edition and Service Pack 2
- Internet Information Server (IIS)
 - NT 4.0 - Option Pack IIS 4.0
 - Windows 2000 - IIS 5.0

Application Server must have:

- Merant Connect JDBC 2.0

Note: Merant is required for the Java Application Server. You may run into a problem if you don't have Java on the machine when installing Merant. You may have to install Merant after you install the TBSM Application Server.

Merant should be installed in the default path. If you install Merant in a location other than the default path, you may need to uncomment the CLASSPATH and update it in the servers.properties file. See the TBSM Administration Guide for more information.

- Internet Information Server (IIS)
 - NT 4.0 - Option Pack IIS 4.0
 - Windows 2000 - IIS 5.0

Propagation Server must have requires no additional software

Event Handler Server for OS/390 must have:

- SQL 7.0 client utilities
- Microsoft Host Integration Server / SNA Server 4.0 and Service Pack 2 Client

SNA Server must have

- Microsoft Host Integration Server, Version 5.0 or SNA Server 4.0 and Service Pack 2
- SQL 7.0 Client utilities

Hardware and Software Requirements for Testing or Quality Assurance

Hardware

Application Server and Propagation Server	Event Handler Server and SNA Server	SQL Server
Single Pentium III, 866MHz	Single Pentium III, 866MHz Xeon (1 MB cache)	Dual Pentium III, 866MHz Xeon (1 MB cache)
512MB RAM	512MB RAM	2GB RAM
System Drive: RAID1, 18GB	System Drive: RAID1, 18GB	System Drive: RAID1, 18GB
Data Drive: RAID1, 18GB	Data Drive: RAID1, 18GB	Data Drive: 50GB

All servers must have:

- Redundant Network Interface Card (NIC)
- Redundant Power Supply

Software

All servers must have:

- Windows 2000 Server and Service Pack 1 or NT Server 4.0 and Service Pack 6a
- MKS Toolkit 7.0
- NT 4.0 Resource Kit Supplement 3

Note: This resource kit is also required on Windows 2000. Do not use Windows 2000 Resource Kit.

- Internet Explorer (IE) Version 5.0

SQL Server must have:

- SQL 7.0 and Service Pack 2
- Microsoft Host Integration Server / SNA Server 4.0 and Service Pack 2 Client

History Server must have:

- SQL 7.0 and Service Pack 2
- Internet Information Server (IIS)
 - NT 4.0 - Option Pack IIS 4.0
 - Windows 2000 - IIS 5.0

Application Server must have:

- Merant Connect JDBC 2.0
- Internet Information Server (IIS)
 - NT 4.0 - Option Pack IIS 4.0
 - Windows 2000 - IIS 5.0

Propagation Server requires no additional software.

Event Handler Server must have

- SQL 7.0 client utilities
- Microsoft Host Integration Server / SNA Server 4.0 and Service Pack 2 Client

SNA Server must have

- Microsoft Host Integration Server, Version 5.0 or SNA Server 4.0 and Service Pack 2
- SQL 7.0 Client utilities

Note: Due to the rapidly changing IT environment, the Hardware and Software requirements for Tivoli Business Systems Manager may change from time to time. These are the minimum requirements.

Hardware and Software Requirements - Workstations

Hardware

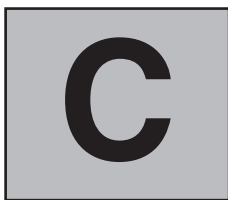
Following are hardware requirements for workstations.

Workstations
Single Pentium III, 500MHz
256MB RAM
100MB free disk space

Software

Following are software requirements for workstations.

Workstations
NT Workstation 4.0 & Service Pack 6a
OR
Windows 2000 Professional and Service Pack 1



Distributed Edition Hardware & Software

The following tables display the hardware and software requirements for TBSM Distributed Edition.

Hardware

The following table displays hardware requirements.

Application Server and Propagation Server	Database Server
Single Pentium III, 866MHz	Dual Pentium III, 866MHz
1GB RAM	2GB RAM
System Drive: RAID1, 18GB	System Drive: RAID1, 18GB
Data Drive: RAID1, 18GB	Data Drive: RAID5, 50GB

Note: If you're running NT 4.0 with a 4GB system drive, TBSM and its prerequisite software should be installed on an alternative drive with 18GB of free space.

All servers must have:

- Redundant Network Interface Card (NIC)
- Redundant Power Supply

Software

All servers must have:

- Windows 2000 Server and Service Pack 1 or NT Server 4.0 and Service Pack 6a
- MKS Toolkit 7.0
- NT 4.0 Resource Kit Supplement 3

Note: This resource kit is also required on Windows 2000. Do not use Windows 2000 Resource Kit.

- Internet Explorer (IE) Version 5.0

SQL Server must have:

- SQL 7.0 and Service Pack 2

History Server must have:

- SQL 7.0 and Service Pack 2

- Internet Information Server (IIS)
 - NT 4.0 - Option Pack IIS 4.0 or
 - Windows 2000 - IIS 5.0

Application Server must have:

- Merant Connect JDBC 2.0

Note: Merant is required for the Java Application Server. You may run into a problem if you don't have Java on the machine when installing Merant. You may have to install Merant after you install the TBSM Application Server.

Merant should be installed in the default path. If you install Merant in a location other than the default path, you may need to uncomment the CLASSPATH and update it in the servers.properties file. See the TBSM administration Guide for more information.

- Internet Information Server (IIS)
 - NT 4.0 - Option Pack IIS 4.0 or
 - Windows 2000 - IIS 5.0

Propagation Server requires no additional software.

TBSM - Distributed Edition Installed Components

The following table displays the installed components for the application and the database server for TBSM - Distributed Edition.

Application Server	Database Server
Application Server Files	SQL Extension Files
Tools and Utilities	Tools and Utilities
Propagation Agent Components	Staged Event Loader
Workstation Program Files (optional)	NT Agent Listener
Help Files (optional)	TBSM xdfparser
	TBSM TDS
	Workstation Program Files (optional)
	Help Files (optional)
	Reporting System (optional)
	Active Documentation (optional)

TBSM Managed Endpoint Software
Operating Systems: (see release notes for version and ptf requirements) AIX, Win NT, Solaris, HP-UX 10.20 or 11.0
Framework: Tivoli Framework Managed Node or TMA endpoint 3.6.1+ Tivoli Distributed Monitoring 3.6.1+ Tivoli ACF (for endpoint support)
Tivoli Management Region (TMR) Server Software: Tivoli Framework 3.6.1+ Tivoli Distributed Monitoring 3.6.1+ Tivoli Software Distribution 3.6.1+



Software Requirements for Non-Windows Platforms

Following are the software requirements for AIX, Linux, and Solaris.

AIX AUX 4.3.3 plus the PTF plus any current patches required to run Java 1.3 on the platform

Linux Linux kernel 2.2.2.x plus any current patches required to run Java 1.3 on the platform

Solaris
Solaris 2.6 or higher plus any current patches required to run Java 1.3 on the platform

Note: To determine the patches required to run Java 1.3, contact the vendor or go to the vendor's Web site for downloading information.



Network Configuration Using Firewalls

To enable communication between TBSM servers and its client environment through a firewall or router, it is necessary to define the ports that are used for communication between TBSM components. Activate the indicated protocols for each port. Depending on the firewall or router configuration, it may be necessary to define ports 135, 136, 137 and 138 for TCP and UDP protocols.

Host	Service	Port	Direction	Notes
TBSM NT Agent – ASINT				
	Tcpmux / TCP Port Service Multiplexer	TCP, UDP 1	Server -> Client	Pinging from Host to specified Server
	Telnet	TCP, UDP 23	Server -> Client	General connectivity
	Domain / Domain Name Server	TCP, UDP 53	Server -> Client	General connectivity
	WWW / World Wide Web HTTP	TCP, UDP 80	Client -> Server	General connectivity
	netbios-ssn / NetBIOS Session Service	TCP, UDP 139	Server -> Client	ASINT Agent
Microsoft SNA Server – SNAServer01				
	ms-sna-server	TCP, UDP 1477	Bi-directional	MVS data feeds
	netbios-ns / NetBIOS Name Service	TCP 137	Bi-directional	MVS data feeds
	netbios-dgm / NetBIOS Datagram Service	TCP 138	Bi-directional	MVS data feeds
	netbios-ssn / NetBIOS Session Service	TCP 139	Bi-directional	MVS data feeds
		TCP 9403	Bi-directional	MVS data feeds
		TCP 1039	Bi-directional	MVS data feeds
	Tcpmux / TCP Port Service Multiplexer	TCP, UDP 1	Bi-directional	Pinging from Host to specified Server
	Telnet	TCP, UDP 23	Bi-directional	General connectivity
	Domain / Domain Name Server	TCP, UDP 53	Bi-directional	General connectivity
	WWW / World Wide Web HTTP	TCP, UDP 80	Bi-directional	General connectivity
	Microsoft SNA Base	TCP, UDP 1478	Bi-directional	MVS data feeds
Domain Controller				
	netbios-ns / NetBIOS Name Service	TCP 137	Bi-directional	MVS data feeds
	netbios-dgm / NetBIOS Datagram Service	TCP 138	Bi-directional	MVS data feeds
	netbios-ssn / NetBIOS Session Service	TCP 139	Bi-directional	MVS data feeds
		TCP 9403	Bi-directional	MVS data feeds
		TCP 1039	Bi-directional	MVS data feeds
	Tcpmux / TCP Port Service Multiplexer	TCP, UDP 1	Bi-directional	Pinging from Host to specified Server
	Telnet	TCP, UDP 23	Bi-directional	General connectivity
	Domain / Domain Name Server	TCP, UDP 53	Bi-directional	General connectivity

Host	Service	Port	Direction	Notes
	WWW / World Wide Web HTTP	TCP, UDP 80	Bi-directional	General connectivity
MQSeries				
		TCP 1411	Bi-directional	For MQSeries Queue Manager feed
		TCP 14xx	Bi-directional	For MQSeries feeds
TBSM Log Forwarder – asisendlog				
		TCP 9400	Server -> Client	For UNIX feeds
TBSM Client Workstation 2.1.2.2+				
		TCP 9401	Client -> Server	Workstation to Application Service
		TCP 9402	Server -> Client	Notification Service to Workstation (MultiCaster)
	courier / RPC	TCP 530	Client -> Server	Workstation to Server connectivity

Configuring TBSM Windows NT Components

This section explains how to modify the Windows NT registry to use fixed ports.

Windows NT Registry Modifications

In a default installation of TBSM, the ports used for the **Workstation to Application Service** and **Notification Service to Workstation** connections are assigned dynamically. To override this behavior and use fixed ports as described in the preceding section, configure the following ports:

- This setting defines the port address to communicate from the client workstation to the Application Service. Update this setting on the Application Service host and any client workstation.

Key	HKEY_LOCAL_MACHINE \ SOFTWARE \ Accessible Software, Inc. \ Access1 \ 1.0 \ Components \ ASIApplicationSvc \ Settings
Name	Port
Type	DWORD
Data	0x000024b9(9401)

- This setting defines the port address to communicate from the Notification Service to the MultiCaster on the client workstation. Update this setting on the Notification Service host and any client workstation.

Key	HKEY_LOCAL_MACHINE \ SOFTWARE \ Accessible Software, Inc. \ Access1 \ 1.0 \ MultiCaster \ Settings
Name	Port
Type	DWORD
Data	0x000024ba(9402)

TBSM SQL Server Configuration

If there is network address translation occurring between the TBSM servers and the client workstations, the following actions must be taken.

On a server running Microsoft SQL Query Analyzer connected to the TBSM Object database, create the records used for the lookup of TCP/IP network address translation (NAT) records. The following SQL query operations define the NAT records, this allows the Notification Service to communicate with the MultiCaster running on the client workstation.

For each unique NAT pair, perform the following SQL command:

```
INSERT INTO network_address_translations
(original_address, mapped_address)
VALUES
('ncacn_ip_tcp:<host>[<port>]', 'ncacn_ip_tcp:<virtual>:[<port>']')
```

In the preceding example the following definitions apply:

Variable Name	Definition
<host>	The actual TCP/IP address of the client workstation as seen on the client workstation.
<virtual>	The TCP/IP address assigned by the translation device (firewall, router, etc.) used to mask the actual address.
<port>	The TCP/IP port used for the Notification Service to MultiCaster communication, typically 9402



Layouts & States for Automation EDI

This Appendix provides application capture modification samples, record layouts, and available states for automation EDI.

Application Capture Modification Samples

This section provides modification samples for application capture.

■ SA OS/390

```
/* Pass MESSAGE to Object */
'GTMEDISA' START_OF_DATA
Token = "GTM" || rc
'GTMEDISA' TOKEN 07 02 /* SA390 MESSAGE EVENT */
'GTMEDISA' TOKEN 05 IPO1 /* SMF ID */
'GTMEDISA' TOKEN 03 CICSREG1 /* OBJECT NAME */
'GTMEDISA' TOKEN 08 SAM001I /* MESSAGE ID */
'GTMEDISA' TOKEN 49 "Test message from SA390" /* MESSAGE TEXT */
'GTMEDISA' TOKEN END_OF_DATA TRACEOFF

/* Pass EXCEPTION to Object */
'GTMEDISA' START_OF_DATA
Token = "GTM" || rc
'GTMEDISA' TOKEN 07 03 /* SA390 EXCEPTION EVENT */
'GTMEDISA' TOKEN 05 IPO1 /* SMF ID */
'GTMEDISA' TOKEN 03 CICSREG2 /* OBJECT NAME */
'GTMEDISA' TOKEN 08 TMAX /* EXCEPTION */
'GTMEDISA' TOKEN 49 500 /* EXCEPTION VALUE */
'GTMEDISA' TOKEN END_OF_DATA TRACEOFF

/* Pass IMS data */
'GTMEDISA' START_OF_DATA
Token = "GTM" || rc
'GTMEDISA' TOKEN 18 03 /* IMS MESSAGE EVENT */
'GTMEDISA' TOKEN 02 IMSS /* IMS OBJECT TYPE */
'GTMEDISA' TOKEN 15 USIBMNT.NTA7MVS /* NETWORK NAME AND VTAM APPL ID */
'GTMEDISA' TOKEN 16 IPO1 /* SMF ID */
'GTMEDISA' TOKEN 17 IMA1CTL /* IMS SUBSYSTEM NAME */
'GTMEDISA' TOKEN 63 THRESHOLD /* STATE OF OBJECT */
'GTMEDISA' TOKEN 64 RM0710W /* MESSAGE ID */
'GTMEDISA' TOKEN 65 "Long message queue > x % ..." /* MESSAGE TEXT */
'GTMEDISA' TOKEN END_OF_DATA TRACEOFF

/* Pass OPC Exception data */
'GTMEDISA' START_OF_DATA
Token = "GTM" || rc
'GTMEDISA' TOKEN 07 05 /* OPC Exception EVENT */
'GTMEDISA' TOKEN 03 OPCTST01 /* OBJECT NAME */
'GTMEDISA' TOKEN 05 IPO1 /* SMF ID */
'GTMEDISA' TOKEN 08 'BATCHREPHR' /* EXCEPTION */
'GTMEDISA' TOKEN 10 OPCTST01 /* OPC APPLICATION NAME */
'GTMEDISA' TOKEN 11 000406005 /* OPC INPUT ARRIVAL TIME */
'GTMEDISA' TOKEN 12 CPU1 /* OPC WORKSTATION ID */
'GTMEDISA' TOKEN 49 'REPLY FOR JOB OPCTST01' /* MESSAGE TEXT */
'GTMEDISA' TOKEN END_OF_DATA TRACEOFF
```

■ OPS/MVS

```
/* Pass MESSAGE to Object */
ADDRESS "TSO" "OI GTMEDIOP" START_OF_DATA
Token = "GTM" || rc
"OI GTMEDIOP" TOKEN 09 02 /* OPSMVS MESSAGE EVENT */
"OI GTMEDIOP" TOKEN 05 IP01 /* SMF ID */
"OI GTMEDIOP" TOKEN 03 CICSREG1 /* OBJECT NAME */
"OI GTMEDIOP" TOKEN 08 SAM001I /* MESSAGE ID */
"OI GTMEDIOP" TOKEN 49 "Test message from OPSMVS" /* MSGTXT */
"OI GTMEDIOP" TOKEN END_OF_DATA TRACEOFF

/* Pass EXCEPTION to Object */
"OI GTMEDIOP" START_OF_DATA
Token = "GTM" || rc
"OI GTMEDIOP" TOKEN 09 03 /* OPSMVS EXCEPTION EVENT */
"OI GTMEDIOP" TOKEN 05 IP01 /* SMF ID */
"OI GTMEDIOP" TOKEN 03 CICSREG2 /* OBJECT NAME */
"OI GTMEDIOP" TOKEN 08 TMAX /* EXCEPTION */
"OI GTMEDIOP" TOKEN 49 500 /* EXCEPTION VALUE */
"OI GTMEDIOP" TOKEN END_OF_DATA TRACEOFF

/* Pass IMS data */
"OI GTMEDIOP" START_OF_DATA
Token = "GTM" || rc
"OI GTMEDIOP" TOKEN 18 03 /* IMS MESSAGE EVENT */
"OI GTMEDIOP" TOKEN 02 IMSS /* IMS OBJECT TYPE */
"OI GTMEDIOP" TOKEN 15 USIBMNT.NTA7MVS /* NETWORK.APPLID */
"OI GTMEDIOP" TOKEN 16 IP01 /* SMF ID */
"OI GTMEDIOP" TOKEN 17 IMA1CTL /* IMS SUBSYSTEM NAME */
"OI GTMEDIOP" TOKEN 63 THRESHOLD /* STATE OF OBJECT */
"OI GTMEDIOP" TOKEN 64 RM0710W /* MESSAGE ID */
"OI GTMEDIOP" TOKEN 65 "Long message queue > x % ..." /*TEXT*/
"OI GTMEDIOP" TOKEN END_OF_DATA TRACEOFF

/* Pass OPC Exception data */
"OI GTMEDIOP" START_OF_DATA
Token = "GTM" || rc
"OI GTMEDIOP" TOKEN 09 05 /* OPC Exception EVENT */
"OI GTMEDIOP" TOKEN 03 OPCTST01 /* OBJECT NAME */
"OI GTMEDIOP" TOKEN 05 IP01 /* SMF ID */
"OI GTMEDIOP" TOKEN 08 'BATCHREPHR' /* EXCEPTION */
"OI GTMEDIOP" TOKEN 10 OPCTST01 /* OPC APPLICATION NAME */
"OI GTMEDIOP" TOKEN 11 000406005 /* OPC INPUT ARRIVAL TIME */
"OI GTMEDIOP" TOKEN 12 CPU1 /* OPC WORKSTATION ID */
"OI GTMEDIOP" TOKEN 49 'REPLY FOR JOB OPCTST01' /* MSG TEXT */
"OI GTMEDIOP" TOKEN END_OF_DATA TRACEOFF
```

■ AF/OPERATOR

```
/* Pass MESSAGE to Object */
CALL "GTMEDIAF" START_OF_DATA
Token = "GTM" || result
CALL "GTMEDIAF" TOKEN 10 02 /* AF/OPER MESSAGE EVENT */
CALL "GTMEDIAF" TOKEN 05 IP01 /* SMF ID */
CALL "GTMEDIAF" TOKEN 03 CICSREG1 /* OBJECT NAME */
CALL "GTMEDIAF" TOKEN 08 SAM001I /* MESSAGE ID */
CALL "GTMEDIAF" TOKEN 49 "Test message from AF Operator" /*MESSAGE TEXT */
CALL "GTMEDIAF" TOKEN END_OF_DATA TRACEOFF

/* Pass EXCEPTION to Object */
CALL "GTMEDIAF" START_OF_DATA
Token = "GTM" || result
CALL "GTMEDIAF" TOKEN 10 03 /* AF/OPERATOR EXCEPTION EVENT */
CALL "GTMEDIAF" TOKEN 05 IP01 /* SMF ID */
CALL "GTMEDIAF" TOKEN 03 CICSREG2 /* OBJECT NAME */
CALL "GTMEDIAF" TOKEN 08 TMAX /* EXCEPTION */
CALL "GTMEDIAF" TOKEN 49 500 /* EXCEPTION VALUE */
CALL "GTMEDIAF" TOKEN END_OF_DATA TRACEOFF
```



```

/* Pass IMS data */
CALL "GTEDIAF" START_OF_DATA
Token = "GTM" || result
CALL "GTEDIAF" TOKEN 18 03 /* IMS MESSAGE EVENT */
CALL "GTEDIAF" TOKEN 02 IMSS /* IMS OBJECT TYPE */
CALL "GTEDIAF" TOKEN 15 USIBMNT.NTA7MVS/* NETWORK NAME AND VTAM APPL ID */
CALL "GTEDIAF" TOKEN 16 IPO1 /* SMF ID */
CALL "GTEDIAF" TOKEN 17 IMA1CTL /* IMS SUBSYSTEM NAME */
CALL "GTEDIAF" TOKEN 63 THRESHOLD /* STATE OF OBJECT */
CALL "GTEDIAF" TOKEN 64 RM0710W /* MESSAGE ID */
CALL "GTEDIAF" TOKEN 65 "Long message queue > x % ..."/* MESSAGE TEXT*/
CALL "GTEDIAF" TOKEN END_OF_DATA TRACEOFF

/* Pass OPC Exception data */
CALL "GTEDIAF" START_OF_DATA
Token = "GTM" || result
CALL "GTEDIAF" TOKEN 10 05 /* OPC Exception EVENT */
CALL "GTEDIAF" TOKEN 03 OPCTST01 /* OBJECT NAME */
CALL "GTEDIAF" TOKEN 05 IPO1 /* SMF ID */
CALL "GTEDIAF" TOKEN 08 'BATCHREPHR' /* EXCEPTION */
CALL "GTEDIAF" TOKEN 10 OPCTST01 /* OPC APPLICATION NAME */
CALL "GTEDIAF" TOKEN 11 000406005 /* OPC INPUT ARRIVAL TIME */
CALL "GTEDIAF" TOKEN 12 CPU1 /* OPC WORKSTATION ID */
CALL "GTEDIAF" TOKEN 49 'REPLY FOR JOB OPCTST01' /* MSG TEXT */
CALL "GTEDIAF" TOKEN END_OF_DATA TRACEOFF

```

■ MAIN VIEW/AUTO OPERATOR

```

/* Pass MESSAGE to Object */
"GTEDIAO" START_OF_DATA
Token = "GTM" || rc
"GTEDIAO" TOKEN 12 02 /* MAINVIEW/AUTO OPERATOR MESSAGE EVENT */
"GTEDIAO" TOKEN 05 IPO1 /* SMF ID */
"GTEDIAO" TOKEN 03 CICSREG1 /* OBJECT NAME */
"GTEDIAO" TOKEN 08 SAM001I /* MESSAGE ID */
"GTEDIAO" TOKEN 49 "Test message from Auto Operator" /* MSGTXT */
"GTEDIAO" TOKEN END_OF_DATA TRACEOFF

/* Pass EXCEPTION to Object */
"GTEDIAO" START_OF_DATA
Token = "GTM" || rc
"GTEDIAO" TOKEN 12 03 /* MAIN VIEW/AUTO OPER EXCEPTION EVENT */
"GTEDIAO" TOKEN 05 IPO1 /* SMF ID */
"GTEDIAO" TOKEN 03 CICSREG2 /* OBJECT NAME */
"GTEDIAO" TOKEN 08 TMAX /* EXCEPTION */
"GTEDIAO" TOKEN 49 500 /* EXCEPTION VALUE */
"GTEDIAO" TOKEN END_OF_DATA TRACEOFF

/* Pass IMS data */
"GTEDIAO" START_OF_DATA
Token = "GTM" || rc
"GTEDIAO" TOKEN 18 03 /* IMS MESSAGE EVENT */
"GTEDIAO" TOKEN 02 IMSS /* IMS OBJECT TYPE */
"GTEDIAO" TOKEN 15 USIBMNT.NTA7MVS/* NETWORK NAME AND VTAM APPL ID */
"GTEDIAO" TOKEN 16 IPO1 /* SMF ID */
"GTEDIAO" TOKEN 17 IMA1CTL /* IMS SUBSYSTEM NAME */
"GTEDIAO" TOKEN 63 THRESHOLD /* STATE OF OBJECT */
"GTEDIAO" TOKEN 64 RM0710W /* MESSAGE ID */
"GTEDIAO" TOKEN 65 "Long message queue > x % ..." /*MESSAGE TEXT */
"GTEDIAO" TOKEN END_OF_DATA TRACEOFF

/* Pass OPC Exception data */
"GTEDIAO" START_OF_DATA
Token = "GTM" || rc
"GTEDIAO" TOKEN 12 05 /* OPC Exception EVENT */
"GTEDIAO" TOKEN 03 OPCTST01 /* OBJECT NAME */
"GTEDIAO" TOKEN 05 IPO1 /* SMF ID */
"GTEDIAO" TOKEN 08 'BATCHREPHR' /* EXCEPTION */
"GTEDIAO" TOKEN 10 OPCTST01 /* OPC APPLICATION NAME */

```

```

"GTMEDIAO" TOKEN 11 000406005 /* OPC INPUT ARRIVAL TIME */
"GTMEDIAO" TOKEN 12 CPU1 /* OPC WORKSTATION ID */
"GTMEDIAO" TOKEN 49 'REPLY FOR JOB OPCTST01' /* MSG TEXT */
"GTMEDIAO" TOKEN END_OF_DATA TRACEOFF

```

Record layout for State Changes, Messages, and Exceptions

Follow matrices below to pass State Changes, Messages, and Exceptions via the REXX EDI.

Format Type = "??" (?? = 07 for SA390, 09 for OPSMVS, 10 for AFOPER, 12 for AUTOOPER) Action Type = "01" (State Change)					
Field Name	Data Type	Value	Field Type	Required	Comment
Object_Name	03	Name of object (STC, Batch job, etc.)	CHAR	Yes	
SMFID	05	Name of MVS Image	CHAR	Yes	
Resource_Type	06	Type of Object (Subsystem)		Yes	
Event_Type	08	State	CHAR	Yes	

Format Type = "??" (?? = 07 for SA390, 09 for OPSMVS, 10 for AFOPER, 12 for AUTOOPER) Action Type = "02" (Message)					
Field Name	Data Type	Value	Field Type	Required	Comment
Object_Name	03	Name of object (STC, Batch job, etc.)	CHAR	Yes	
SMFID	05	Name of MVS Image	CHAR	Yes	
Resource_Type	06	Type of Object (Subsystem)			
Event_Type	08	Message ID	CHAR	Yes	
Event_State	09	State	CHAR		Use if passing State Change from message text.
Event_Test	49	Message text to be passed	CHAR	Yes	

Format Type = "??" (?? = 07 for SA390, 09 for OPSMVS, 10 for AFOPER, 12 for AUTOOPER) Action Type = "03" (Exception)					
Field Name	Data Type	Value	Field Type	Required	Comment
Object_Name	03	Name of object (STC, Batch job, etc.)	CHAR	Yes	
SMFID	05	Name of MVS Image	CHAR	Yes	
Resource_Type	06	Type of Object (Subsystem)			
Event_Type	08	Exception	CHAR	Yes	Exception (TMAX, TMIN, etc.)
Event_State	09	State	CHAR		Use if passing State Change from message text.
Event_Text	49	Message text to be passed	CHAR	Yes	Value of exception (50, 100, etc.)

Format Type = "??" (?? = 07 for SA390, 09 for OPSMVS, 10 for AFOPER, 12 for AUTOOPER) Action Type = "04" (OPC Message)					
Field Name	Data Type	Value	Field Type	Required	Comment
Object_Name	03	Name of object (STC, Batch job, etc.)	CHAR	Yes	
SMFID	05	Name of MVS Image	CHAR	Yes	
Resource_Type	06	Type of Object (Subsystem)			
Event_Type	08	State, Message ID, Exception	CHAR	Yes	
Event_State	09	State	CHAR		Use if passing State Change from message text.
APPLID	10	OPC Application ID	CHAR	Yes	Required if passing OPC data
IA	11	OPC Input Arrival Time	CHAR	Yes	Required if passing OPC data
WKSTNID	12	OPC Work Station ID	CHAR	Yes	Required if passing OPC data
Event_Text	49	Message Text to be passed	CHAR	Yes	

Format Type = "??" (?? = 07 for SA390, 09 for OPSMVS, 10 for AFOPER, 12 for AUTOOPER) Action Type = "05" (OPC Exception)					
Field Name	Data Type	Value	Field Type	Required	Comment
Object_Name	03	Name of object (STC, Batch job, etc.)	CHAR	Yes	

SMFID	05	Name of MVS Image	CHAR	Yes	
Resource_Type	06	Type of Object (Subsystem)			
Event_Type	08	State, Message ID, Exception	CHAR	Yes	
Event_State	09	State	CHAR		Use if passing State Change from message text.
APPLID	10	OPC Application ID	CHAR	Yes	Required if passing OPC data
IA	11	OPC Input Arrival Time	CHAR	Yes	Required if passing OPC data
WKSTNID	12	OPC Work Station ID	CHAR	Yes	Required if passing OPC data
Event_Text	49	Message Text to be passed	CHAR	Yes	

Available States for Automation EDI

The following states can be passed to TBSM using the REXX EDI.

TBSM	TBSM Alert State	TBSM Priority
DOWN	Green	High
UP	Green	High
CTLDOWN	Green	High
AUTODOWN	Green	High
ENDED	Green	High
INACTIVE	Yellow	High
STARTED2	Yellow	High
HALTED	Yellow	High
STUCK	Yellow	High
ZOMBIE	Yellow	High
BROKEN	Red	High
STOPPED	Green	High



SA/390, OPS/MVS, and AF/Oper EDI Variables

The following variables were used for the automation EDI REXX programs 'AOPEDISA', 'AOPEDIOP' and 'AOPEDIAF'. These programs have been enhanced and renamed to GTMEDIXX. This appendix is provided for those using the older code.

traceopt

This parameter provides for diagnostic functionality as to the operations performed in support of the passing of data.

Attribute values for this parameter are:

- TRACEOFF
- TRACEON

format_type

This parameter references the source of the data that is to be passed.

Values for this parameter are:

- SA390
- OPSMVS
- AFOPER

action_type

This parameter provides the type of action that was initiated by the product or application.

Values for this parameter are:

- STATECHANGE
- MESSAGE
- EXCEPTION

smfid

This parameter contains the value of the system ID or system name from where the message or exception was generated.

object_type

This parameter provides the Object Type of the enterprise component that is monitored by the program or application that generated the message or exception.

object_name

This parameter contains the value of the actual name of the object that generated the message or exception.

state_msgid_excpid

This parameter contains the value of the actual Message ID or Exception ID. In the case of an action_type of STATECHANGE, the Object Type is the value of this parameter.

msg_state

This parameter contains the state of the object from the message text.

msg_text

The actual text associated with any message or exception is placed within this parameter.

application_id

This parameter contains the OPC application name.

input_arrival_time

This parameter contains the OPC arrival time for a specific operation.

work_station

This parameter contains the name of the OPC workstation associated with the application id.

Note: Parameter values shown in ALL CAPITALS must be enclosed in double quotes (“MESSAGE”). In the event that any of the parameters do not have a valid value assigned by the message or exception, the value of that parameter should be set to “NULL”



Installing/Configuring SNA Server

This appendix describes how to install and configure Microsoft SNA Server so that it communicates with the Object Server component of Source/390.

SNA Client/Server Security

SNA client applications access the mainframe Object Server task via the SNA server machine. For this reason, the SNA client requires authority to access the SNA server machine and vice versa. For this to be accomplished, the following important NT security issues must be addressed:

- The SNA server and the SNA client must be installed with the same userid and password.
- If you install the SNA server and the SNA client with a userid that already exists, you must manually give that userid NT authority to Act as part of the Operating System.
- If you logon to the SNA client machine with a userid and password that is different from what you used to install the SNA software, this userid and password must exist with the same password on the SNA server machine.

Installing the SNA Server

Note: Microsoft SNA Server Version 4, Service Pack 2 or above is required.

To install the SNA server:

1. Run the Setup utility that is located in the I386 directory of the SNA Server installation drive.
2. When Setup is launched, read the Welcome screen and follow the instructions.
3. Follow the instructions on each screen as you proceed through the installation.

Use the default options throughout the installation. Following is additional information:

- In the **Choose Server Role** dialog, choose **Primary configuration server**, as the primary SNA server maintains the master copy of the configuration file.
- In the Network **Subdomain Name** dialog, the subdomain name must be unique. The subdomain name is the NT domain in which the SNA server is installed.
- In the **Server Domain Account Information** dialog, the password is case-sensitive.
- In the **New Client Access License** dialog, choose a licensing mode. Generally, the **Per Server** node is used. Ensure the **Quantity** field has at least 2 client licenses. The quantity determines the number of concurrent client connections.

-
- When the installation is complete, restart the computer. When the computer comes back up, update the SNAServer service, so that it starts automatically. Start the service.

Configuring the SNA Server

SNA Server Manager is the managing component within a Windows NT domain that sponsors sessions on behalf of users between SNA server and VTAM on the mainframe. If you have questions, many answers to configuring SNA Manager can be found in the help facilities provided by SNA Server.

Before getting started, work with your VTAM systems programmer to complete the following chart. Many of these values are required during the configuration of SNA server and client.

Parameter	Value*	Description
VTAM Major Node Names		Applid and PU major node names
Network Name		NETID(D NET,VTAMOPTS,OPTION=NETID)
Control Point Name		CPNAME in PU major node
PU Name		name of PU as defined to VTAM
Local Node ID(XID)		IDNUM/IDBLK from PU major node**
Local/Partner LU alias/LU name		LU with LOCADDR=0 in PU major node
Remote LU alias/LU name		Applid name for TBSM (one for each host)
Remote Network Address		MAC address of network interface card**
Remote SAP address		SAPADDR of mainframe network interface (default is 4)**
SNA Server Machine Name		name of NT machine where SNA Server is installed
* Indicates that these values are entered site specific		
** Indicates parameters for switched connections		

Install Microsoft's SNA Server on the Windows NT workstation that is physically connected with a Token Ring or Ethernet card. This workstation should be locally or remotely attached to MVS.

The following steps identify the tasks involved in configuring the SNA Server sections. SNA Server is designed like Windows Explorer, using a tree-like hierarchy. Each section requires an insertion of an object type that is to be created and configured.

A token ring network configuration is provided here to show a sample of how to configure the SNA Server. (For additional information, refer to the SNA Server help.)

The tree structure of SNA Server that is to be configured is:

Servers

Link Services

SNA Service

Connections

Local APPC LUs

Remote APPC LUs

APPC Modes

CPIC Symbolic Names

Under servers, locate the name of the server in which SNA server is installed and make the following configuration changes. SNA Server is a traditional Windows application and you can access the properties menu by right clicking any object and then selecting the properties button.

1. **Insert a SNA Service.** You are required to provide the Network Name and Control Point Name of the VTAM environment to which the SNA manager will be connected. After inserting the SNA Service, you must edit its properties to add the Network Name and Control Point Name.

Title	Name for SNA Service
Comment is optional	
Network Name is required	
Control Point Name is required	

2. **Insert a Link Service** on behalf of the SNA Service created using the protocol that supports the type of physical connection being used. For example, Token Ring can use the DLC 802.2 protocol.

(The protocol used must be installed on the machine. If it is not, you will get an error message indicating that the protocol is not installed on the computer. If the protocol is not installed on the computer, you may add it from the Windows NT Control panel by selecting the Network icon and then selecting the Protocols tab. You will need the Windows NT installation disk to add the protocol. After you add the protocol, reboot the computer before continuing the SNA server configuration.)

Title	Name for Link Service
Adapter name of Token Ring	
SAP	x04
Check	Fixed SAP
Check	Allow Link Service to be distributed

3. **Insert a Connection for the Link Service.**

■ General Tab

Title	Name for Connection Service
Name	Name for connection (PU name is often a useful name here).
Select service name of link service	
Comment is optional	
Remote End	Host System

Title	Name for Connection Service
Allowed Directions	Both Directions
Activation	On Server Startup

■ Address Tab

Title	Name for Connection Service
Remote Network Address	12 Byte Mac Address
Remote SAP address	Remote SAP address from VTAM

■ System Identification Tab

Title	Name for Connection Service
Local Node Name	
Network Name	See #1, Insert a SNA Service
Control Point Name	Cpname defined on the PU2.1 node
Local Node ID	IDNUM and IDBLK defined on the PU2.1 node
XID type	Format 3
Remote Node Name options	Do not fill in
Compression Type	None

■ 802.2 DLC Tab

The default settings are sufficient.

At this point, if the physical connection is set up and the definitions within SNA Server are correct, the connection should activate upon starting the SNA Service.

There is an 'Active' status displayed on the SNA Service and Connection name when started.

4. Insert a Local APPC LU

■ General Tab

Title	Name for Local APPC LU
LU Alias	The name of the Independent LU
Network Name	same name as provided in #1
LU Name	same name as LU Alias
Comment is optional	

■ Advanced Tab

Title	Name for Local APPC LU
Member of default outgoing local APPC LU Pool	Check
Timeout for starting invokable TPs	600 seconds
Implicit Remote LU	None
LU6.2 Type	Independent
SyncPoint Support enabled	Do not check

There is only one Local APPC LU defining the PU2.1 node residing in VTAM.

5. Insert a Remote APPC LU

■ General Tab

Title	Name for Remote APPC LU
Connection	Assign connection name to be used
LU Alias	Remote APPC LU
Network Name	same name as provided in #1
LU Name	same name as LU Alias
Uninterpreted Name	same name as LU Alias
Comment is optional	

■ Options Tab

Title	Name for Remote APPC LU
Parallel Sessions	Check
Implicit Incoming Mode	None
Session level security	No

Setup one Remote APPC LU definition for each APPC applid defined for the OS/390 server address spaces. If there are 8 MVS systems in which TBSM for Windows NT is to collect data from, then there should be 8 Remote APPC LU definitions defined within this section.

6. Insert a APPC Mode – insert the mode(s) name defined in VTAM

■ General Tab

Title	Name for Modetab
Mode Name	LU62PS
Comment is optional	

■ Limits Tab – Must match the VTAM applid definitions

Title	Number
Parallel Session Limit	4
Minimum Contention Winner Limit	1
Partner Min Contention Winner Limit	1
Automatic Activation Limit	0

■ Characteristics Tab – RU sizes must match the VTAM mode table

Title	Number
Pacing Send count	7
Pacing Receive count	7
Max Send RU size	1024
Max Receive RU size	1024

- **Partners Tab**

Follow the Add instructions to associate the LU pairs, Local and Remote LU names.

- **Compression Tab**

The default settings are sufficient.

7. Insert a CPIC Symbolic Name

There are two symbolic names to be inserted. The name ACC1RCV is associated with the process name for data, which is received by the client on behalf of SNA Manager.

The name ACC1RECV is associated with the process name for data sent by the client to OS/390. ACC1RECV is only used for the APPC Upload function. Clients using the FTP upload function do not require ACC1RECV.

ACC1RCV

- **General Tab**

Title	Name for CPIC Symbolic name
Name	ACC1RCV
Comment is optional	
Conversation Security	None
Mode Name	LU62PS

- **Partner Information Tab**

Title	Name for CPIC Symbolic name
Check Application TP	Enter ACC1RCV in the box to the right
Partner LU Name Alias	Local APPC LU name

ACC1RECV

- **General Tab**

Title	Name for CPIC Symbolic name
Name	ACC1RECV
Comment is optional	
Conversation Security	None
Mode Name	LU62PS

- **Partner Information Tab**

Title	Name for CPIC Symbolic name
Check Application TP	Enter ACC1RECV in the box to the right.
Partner LU Name Alias	Remote APPC LU name

SNA Server Manager is now configured and ready to be used by a SNA Manager Client. When making changes to SNA Server, the connection and SNA Service must be stopped. After the changes are made, the configuration must be saved before restarting the SNA service and connection name.

Installing SNA Client

This section describes how to install the Microsoft SNA Client.

1. Run the **Setup** utility located in the Clients\Winnt\I386 directory of the SNA Server installation drive. When Setup is launched, read the **Welcome** screen and follow the instructions.
2. Follow the instructions on each screen as you proceed through the installation.

Use the default options throughout the installation. Following is additional information:

- From the **Options** list menu, select **SNA Server Manager** and **3270 Applet**.
- On the **Client Mode** dialog, select **Client locates servers by name**.
- On the **Remote Server Names** dialog, enter the name of your SNA server.
- Restart Windows to complete the installation of the SNA client.

Configuring the SNA Client

Please refer to Chapter 7, section titled, *Configuring the HIS Client* and use the same information for configuring the SNA client.

Index

Numerics

- 3270 pool terminals 50
- 3270 pool terminals, LOGMODE entries 51

A

- address spaces, Source/390 46
- ANSI NULLs 9
- application server (distributed) install 22
- application server (enterprise), install 14
- applid, VTAM 49
- architecture, distributed 5
- architecture, Source/390 object pump 47
- ASIMVSIPLListener 88
- automation, EDI 72
- availability reporting system 4

B

- base services, defined 7
- batch schedules 46
- bulk discovery 87

C

- cache size, event 40
- CD, distributed edition ix
- CD documentation ix
- CICS, OMEGAMON for 45
- CICS TDQ message file 45
- CICS transient data exit program 67
- CICSplex SM instrumentation 32
- CICSplexSM 46
- client-side options, turn off 9
- client workstation (enterprise), install 17
- command line, install distributed components from 28
- commands, object pump 84
- compatibility services 15, 23
- components, enterprise edition 45
- components, Source/390 45
- configuration options, event handler 100
- configure, networkfirewall, configure network 115, 116
- configure connection to TEC 30
- configure event enablement 33, 38
- configure mainframe components 49
- configure monitoring of instrumentation 43
- configure multiple Tivoli object servers 35
- configure properties file 37

- configure reporting system (distributed) 21
- configure reporting system (enterprise) 13
- configure SNA client 133
- configure SNA server 94, 128
- configure task server 33, 35
- configure TCP/IP port numbers 34
- console monitoring 45
- continuing parameters 53
- control cards, TCP/IP 55

D

- database server (distributed) install 19
- database server (enterprise), install 10
- dataspace, Source/390 46, 48
- dataspace startup parameters 54
- dataspace size 54
- DB2 & DB2PM 46
- DB2, OMEGAMON for 45
- DB2 for OS/390 instrumentation 32
- DB2PM 46
- define logical unit name 37
- definitions, VTAM 49
- desktop 26
- distributed architecture 5
- distributed edition, install 19
- distributed edition CD ix
- distributed edition NT servers 106
- distributed monitoring (GEM) 25
- documentation CD ix

E

- EDI 68
- enterprise edition, install 10
- enterprise edition components 45
- enterprise edition hardware requirements 107
- enterprise edition software requirements 107
- event cache size 40
- event enablement 5
- event enablement, configure 38
- event enablement, event classes 38
- event enablement, start 39
- event enablement, TEC rules 38
- event enablement, configure 33
- event enablement security 33
- event enablement user ID 38
- event handler configuration options 100
- event handlers (enterprise), install 17
- existing GEM customers 25

G

GEM customers 25
GTMAOPE0 87

H

hardware 105
hardware requirements, enterprise edition 107
history server, install 12

I

IMS 46
install, SQL database server 8
install application server (distributed) 22
install application server (enterprise) 14
install client workstation (enterprise) 17
install database server (distributed) 19
install database server (enterprise) 10
install distributed components from command line 28
install distributed edition 19
install enterprise edition 10
install event handlers, (enterprise) 17
install from Tivoli desktop 26
install history server, (enterprise) 12
install LU6.2 support 90
install Omegamon interface 79
install on TEC server 25
install propagation server (enterprise) 16
install SNA client 99, 133
install SNA server 127
install SQL server service pack 8
installed components, distributed edition 112
instrumentation 6, 31
instrumentation, CICSplex 32
instrumentation, configure monitoring of 43
instrumentation, DB2 for OS/390 32
instrumentation, Tivoli NetView for OS/390 32
instrumentation service, monitor 40

J

JES3 53

L

Line of Business 1
LOB. see line of business 1
logical unit name, define 37
LOGMODE entries 51
LOGSCREENS 81
LU. see logical unit 37

LU6.2, install support 90
LU6.2 support, validate 101

M

mainframe components, configure 49
managed nodes, configure for 41
management, object state 3
management server, TBSM 4
max threads 86
message file, CICS TDQ 45
Microsoft SNA server 93, 127
mode table 49
modify command, object server 80
modify commands, object pump 81
MQSeries 32
MVS, OMEGAMON for 45

N

NAT IP 58
NetView/390 51
node PU2.1 90
nodes, managed, configure for 41
NT workstations 106
NTservers, distributed edition 106

O

object pump, Source/390 46
object pump commands 84
object pump modify commands 81
object pump startup parameters 53
object server, Source/390 46, 48
object server modify command 80
object server startup parameters 55
object servers, configure 35
object state management 3
Omegamon, define pool of terminals 50
OMEGAMON for CICS 45
OMEGAMON for DB2 45
OMEGAMON for MVS 45
Omegamon interface, install 79
Omegamon requirements 79
OMVS 60
OPC. see Tivoli workload scheduler 46
operating systems, adding additional 102
operations planning and control 68
OPERCMDS 60
opqueuesize 54
OPS/MVS 46
options 32
OS/390 tasks, set user ID 36
oservs. see object servers 35
osqueuesize 54

P

- packaging ix
- parameter 62
- parameters, continuing 53
- parameters, dataspace startup 54
- parameters, object pump startup 53
- parameters, object server startup 55
- password, SQL authentication 9
- pool terminals, 3270 50
- port numbers, configure 34
- PPI DISABLE 83
- PPI ENABLE 83
- PPI RESET 84
- PPI STATUS 83
- PPI TRACE 83
- propagation agent server (enterprise), install 16
- properties file, configure 37
- PU2.1 node 90

R

- REFRESH 81
- report system, availability 4
- reporting system (distributed) configure 21
- reporting system (enterprise), configure 13
- requirements, Omegamon 79
- resource measurement facility 46
- restart servers (distributed) 24
- restart servers (enterprise) 19
- RMF. *see* resource measurement facility 46
- RODM 83

S

- SA/390 46
- security, event enablement 33
- security, SNA client/server 93, 127
- security, Source/390 60
- security, task server 33
- server, SNA 93, 127
- servers (distributed) restart 24
- servers (enterprise), restart 19
- service desk 4
- set task server user ID 36
- set user ID for OS/390 tasks 36
- setup 7
- SHOW 81
- SHOW COUNTS 85
- SHOW EXCPS 84
- SHOW LOGONS 86
- SHOW MSGCOUNT 83, 85
- SHOW POOLS 82, 85
- SHOW PPICOUNT 83
- SHOW SESSIONS 82, 85
- SHOW TRAPS 82, 84
- SHOW VARS 82, 85

- SNA client, configure 133
- SNA client, install 99, 133
- SNA client/server security 93, 127
- SNA server 93, 127
- SNA server, configure 94, 128
- SNA server, install 127
- software requirements, enterprise edition 107
- Source/390 3, 45
- Source/390 address spaces 46
- Source/390 components 45
- Source/390 dataspace 46, 48
- Source/390 dataspace startup parameters 54
- Source/390 functions 45
- Source/390 object pump 46
- Source/390 object pump architecture 47
- Source/390 object server 46, 48
- Source/390 on MVS image 53
- Source/390 security 60
- SQL Database Server 4
- SQL database server, install 8
- SQL server authentication password 9
- SQL service pack, install 8
- start event enablement 39
- Start Task JCL, Tivoli NetViewTivoli NetView Start Task JCL 52
- start task server 37
- startup parameters, dataspace 54
- system variables 54

T

- table, mode 49
- task server 5
- task server, configure 33, 35
- task server, start 37
- task server, user ID 36
- task server security 33
- TBSM 1.1.1 15, 23
- TBSM EDI 68
- TBSM Management Server 4
- TBSM workstation 4
- TCP/IP, configure port numbers 34
- TEC, configure connection 30
- TEC, configure for non-local 41
- TEC. *see* Tivoli Enterprise Console 5
- TEC rules and event classes, update 38
- TEC server, install components on 25
- TECs, configure support for multiple 41
- threads, max 86
- Tivoli BSM 1.1.1 application server 15, 23
- Tivoli BSM 1.1.1 notification server 15, 23
- Tivoli desktop 26
- Tivoli Enterprise Console 5
- Tivoli Framework 5
- Tivoli instrumentation, configure monitoring of 43
- Tivoli NetView for OS/390 instrumentation 32
- Tivoli OPC 68
- Tivoli Source/390 3
- Tivoli workload scheduler 46

TMA endpoint, configure for 41
TME tasks 38
TMF, see Tivoli Framework 5
TRAPS 82

U

user ID, event enablement 38
user ID, OS/390 tasks 36
user ID, task server 36

V

VTAM applid 49
VTAM definitions 49

W

workload scheduler 46
workstation, client (enterprise), install 17
workstation, TBSM 4
workstations, NT 106



Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.