Tivoli Business Systems Manager

RODM Release Notes

Tivoli Business Systems Manager
RODM Release Notes

Tivoli Business Systems Manager RODM Release Notes

Copyright Notice

Trademarks

Notices

Contents

Chapter 1. Overview of Monitoring RODM
Chapter 1. Overview of Monitoring RODM

1

Overview of Monitoring RODM Resources

The Systems Network Architecture (SNA) provides a centralized network management system for subarea and Advanced Peer-to-Peer Networking (APPN) networks. The Systems Network Topology Manager uses Resource Object Data Manager (RODM) as a data cache, to locate and manage SNA topology data.

RODM is an object-oriented data cache. Objects in RODM represent resources within your SNA network. The data cache, which contains all of the objects, is located entirely in the memory of the host processor resulting in fast access to data and high transaction rates.

The integration of Tivoli Business Systems Manager (TBSM) with RODM enables you to manage system resources such as Customer Information Control System (CICS), DB2 and Batch Schedules together with SNA/APPN-based resources in a single view.This provides significant assistance in visualizing the impact a network resource outage or performance degradation has on your enterprise.

TBSM provides the following functionality for monitoring resources that are stored in RODM:

- Representations of SNA/APPN network resources.
- Discovery of resources that are managed by RODM via user-schedule discovery tasks.

- Monitoring of state changes on RODM resources for a user-selectable subset of objects.
- Configured mapping of RODM state changes to TBSM events.

Software Requirements

- Multiple Virtual Storage (MVS) OS/390 1.3 or higher.
- Transmission Control Protocol/Internet Protocol (TCP/IP) Version 3.2 or higher.
- OS/390 TBSM Version 1.1 plus APAR's OW47080 and OW48019 or TBSM version 1.5.
- NetView Version 1.2 or higher with PTF's UW76035(V1.2) or UW76034(V1.3) to add the notify method authorized program analysis report (APAR) OW45746 for 1.2 and 1.3.
- NT TBSM Version 1.1.1 with patch TBSM 1.1.1-BSM-0001 or NT TBSM Version 1.5.

Chapter 2. Discovery Processing
Chapter 2. Discovery Processing

2

Discovery Processing

All networks are dynamic. The network object discovery process and subsequent registration ensure that the TBSM model is up-to-date and accurately reflects the topology of your network. When RODM restarts, the SNA Topology Manager objects may change.Therefore, when a RODM restart is detected, you may perform an unload of the RODM data cache and a reload of the TBSM database. You perform the discovery process and the registration process whenever definition changes have been made.

SNA/APPN resources managed by RODM are discovered via a process involving the following steps:

1. Unload of RODM object cache to datasets on OS/390.
2. Transmission of datasets to TBSM NT-based server.
3. Loading of data into TBSM database.
4. Processing of data to perform mapping between RODM objects and TBSM object database.

You run the preceding steps separately; as a manual or automated process.

The discovery process handles initial discovery scenarios and rediscovery scenarios. In initial discovery, no previously discovered RODM objects exist in the TBSM database. All subsequent discoveries are considered rediscoveries, and the discovery process makes the appropriate insertions and updates into the TBSM database to reflect the newest discovery data. SNA/APPN objects, within TBSM that are no

longer present in the RODM database and not rediscovered, are ***not*** automatically deleted within TBSM; this is because a given object not appearing in a subsequent discovery for reasons other than having been deleted. Rather, TBSM ages these objects so that you control when an object, which has not been rediscovered, should be deleted from the TBSM database.

RODM Discovery

The objects you monitor change; SNA devices, IBM Network Control Programs (NCPs), links, and connections are added and deleted. To insure that TBSM accurately reflects the network and all components, discovery of the RODM data cache by TBSM is necessary; typically after scheduled maintenance of the network.

RODM discovery is a batch process, which extracts data from the RODM data cache using the EKGKUNLD utility, a component of RODM, and forwards the data to the GTMAOPE0 utility. The GTMAOPE0 transmits the RODM data down the Transmission Control Protocol/Internet Protocol (TCP/IP) connector, providing the most up-to-date model of the network for assimilation into TBSM.

Once discovered, you must select an object set for registration, in order to monitor (registration is explained later in this chapter).

The following process describes RODM Discovery:

1. Run the EKGKUNLD utility to extract the RODM data.
2. Run the GTMAOPE0 Internet Protocol (IP) utility, which transmits singular RODM data to TBSM.

Note:

- The EKGKUNLD utility documentation is found in the *TME bookshelves for NetView for OS/390.*
- Refer to the GTMAOPE0 documentation in Appendix C on page 49 on how to setup and use the IP utility.
- Refer to the SGTMSAMP library for the RODM Discovery JCL.

Chapter 2. Discovery Processing
Chapter 2. Discovery Processing

Discovery Architecture

The following architectural diagram exhibits the detection and capture of RODM resource data, which loads the TBSM database with SNA network resources.

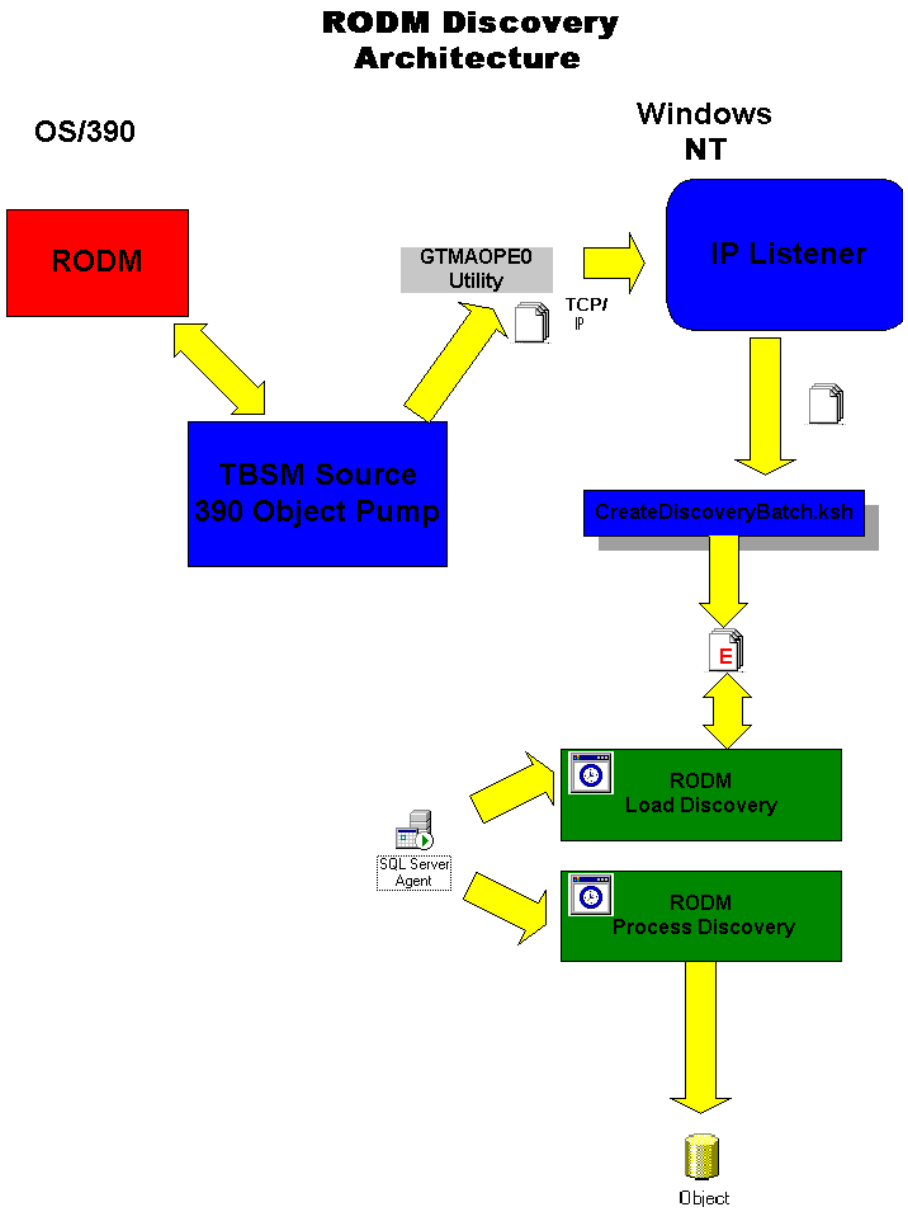


Figure 1. RODM Discovery Architecture

Discovery Flow

- The TBSM Source/390 Object Pump collects resource data from RODM and calls the GTMAOPE0 utility.
- The GTMAOPE0 utility, which is a standalone MVS TCP/IP application, sends the RODM data to the TBSM server running the ASIMVSIPListenerSvc.
- The ASIMVSIPListenerSvc calls the *CreateDiscoveryBatch.ksh*, which is configured in the Registry of the TBSM server that runs the ASIMVSIPListenerSvc.
- The *CreateDiscoveryBatch.ksh* script defined in the Registry creates the *Discovery Batch* and assigns it a state of ENQUEUED in the Structured Query Language (SQL) database.
- *RODM Discovery Load* SQL Server job takes any *Discovery Batch* that is in the ENQUEUED State and loads the associated file into the SQL Server database.
- *RODM Discovery Process* a SQL Server job performs the discovery processing required on any *RODM Discovery Batch* that is in the LOADED state.
- The RODM Discovery Process job updates TBSM object hierarchy.

Registration

When an object is registered, a registration message is sent to the TBSM Source/390 Object Pump running on the MVS system that is connected to RODM, which contains the monitored network topology and model. The information specifying whether an object is registered is stored in the TBSM object attribute *RegistrationStatus*.

RODM Restart

RODM does not need to be up and running when Source/390 (a suite of OS/390 based components of TBSM) starts. If RODM is started after Source/390, Source/390 connects at that time. RODM may be stopped and restarted in which case Source/390 reconnects when RODM starts.

When a RODM restart is detected, registration is sent from the TBSM object database to the RODM Manager. If RODM is cold started or if network changes have occurred, an unload of the RODM database should be performed and reloaded into the TBSM database.

Therefore, when a RODM Stop or Start is detected, a message is sent to TBSM indicating that RODM has stopped or started.This may be used to turn an Operating System running RODM object Red/Green, indicating availability of network monitoring and schedule a RODM unload and load the TBSM database.

3

Object Model

Each object class as defined to RODM is specified by a class name and an optional role name, which is used to indicate a specialized use of the class. Each unique class or role combination is mapped to a TBSM class.

The following table illustrates the mapping for all RODM classes monitored by TBSM:

RODM Class	RODM Role	TBSM Class ID	TBSM Class Name
aggregateGraph2	nnDomain	NNDM	nnDomain
aggregateGraph2	nnDomainNetwork	NNDN	nnDomainNetwork
aggregateGraph2	nnDomainNetworkCluster	NNDC	nnDomainNetwork Cluster
appnBrNN		BRNN	appnBrNN
appnEN		APEN	appnEN
appnNN		APNN	appnNN
appnTransmissionGroup		APTG	appnTG
appnTransmissionGroupCircuit		ATGC	appnTGCircuit
appnTransmissionGroupCircuit	appnTransGroupCircuitCN	TGCN	appnTGCircuitCN

RODM Class	RODM Role	TBSM Class ID	TBSM Class Name
appnTransmissionGroupCircuit	ntriTypeAppnTgCircuit	NTRI	ntriTypeAppnTgCircuit
circuit2	interdomainCircuit	IDCT	IDCircuit
circuit2	interdomainNetworkCircuit	IDNC	IDNetworkCircuit
crossDomainResource		XDR	xDomainResource
crossDomainResourceManager		XDRM	xDomainResourceManager
definitionGroup		DG	definitionGroup
interchangeNode		ICND	interchangeNode
lenNode		LENN	lenNode
logicalLink		LL	logicalLink
logicalUnit		LU	logicalUnit
luGroup		LUG	luGroup
migrationDataHost		MDH	migrationDataHost
port		DLCP	DLCPort
snaLocalTopo	enLocalTopology	ENLT	enLocalTopology
snaLocalTopo	nnLocalTopology	NNLT	nnLocalTopology
snaNode		SNAN	snaNode
subareaTransmissionGroupCircuit		STGC	subareaTGCircuit
t2-1Node		T21N	t2_1Node
t4Node		T4N	t4Node
t4Node	t4NodeGateway	T4NG	t4NodeGateway
t5Node		T5N	t5Node
virtualRoutingNode		VRN	virtualRoutingNode

A description of each class and the accompanying icon appears in Appendix A on page 35.

Class Hierarchy

All RODM classes represented in TBSM are based on the abstract base class *RODMManagedObject.* This class encapsulates behavior that is common to all underlying RODM classes.

The TBSM object classes that derive directly from *RODMManagedObject* class are the following:

- IDCircuit
- IDNetworkCircuit
- enLocalTopology
- luGroup
- nnDomain
- nnDomainNetwork
- nnDomainNetworkCluster
- nnLocalTopology
- subareaTGCircuit

RODMStates Object

The *RODMStatesObject* is derived from *RODMManagedObject* and adds attributes that are based on the RODM states field. The states fields are a vector of bit values that map to seven distinct state indicators.

For more details, refer to the following publication:

TME 10 NetView for OS/390

Data Model Reference

Document Number SC31-8232

SNA/APPN Network Object Hierarchy

TBSM defines the SNA/APPN Network object class, which is used as the aggregation point for all SNA/APPN resources with a given network ID.

Note: SNA/APPN Network objects ***must*** be defined manually in order for resources from that network to be discovered and monitored. The name of this object ***must*** be exactly the name of the Network ID.

The objects classes are arranged in a hierarchy according to the discovered relationships between the objects. These relationships are specified in the link data that is extracted from the RODM database. All SNA Topology Manager classes are rooted under an object of the SNA/APPN Network Object Class, with several, higher-level object classes being directly contained under the network object class.

They are the following:

- enLocalTopology
- interchangeNode
- migrationDataHost
- nnDomain
- nnDomainNetwork
- nnDomainNeworkCluster
- nnLocalTopology
- t4Node
- t4NodeGateway

The following illustration demonstrates a typical hierarchy of RODM objects as they appear in HyperView.

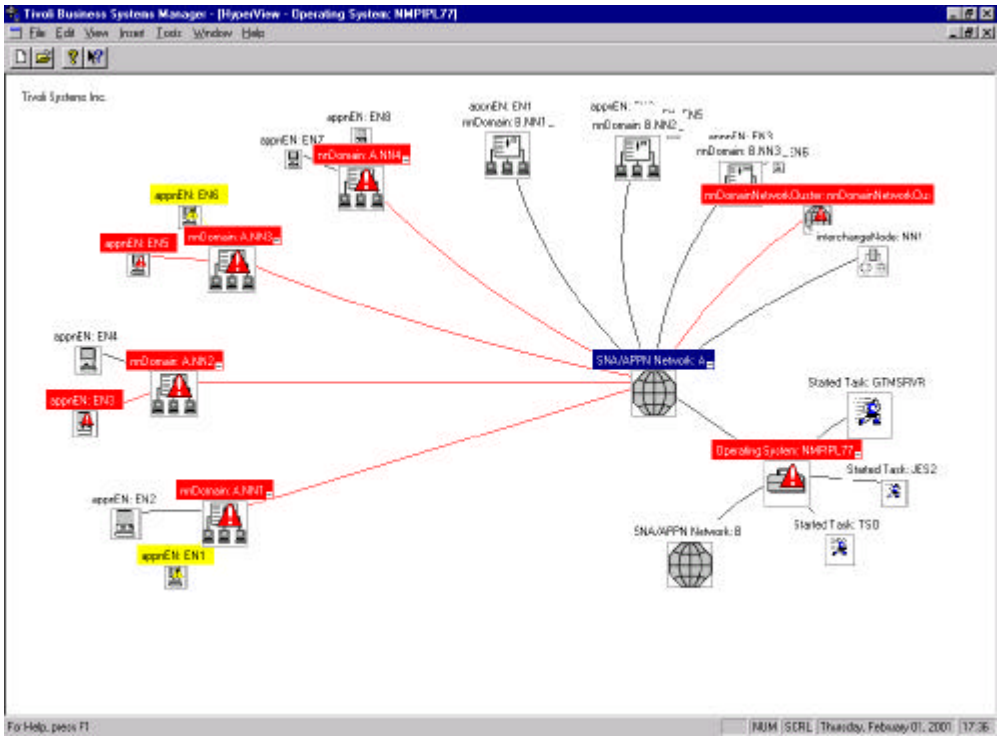


Figure 2. RODM objects within HyperView

Chapter 4. Event Processing
Chapter 4. Event Processing

4

Event Processing

After discovery, the TBSM object database is populated with discovered network resources. In order for these resources to be monitored by TBSM, you must select the set of these resources for which monitoring is required. These resources are considered to be registered for monitoring. All others are unregistered and are not monitored.

Event processing involves capturing specific events and routing them into the TBSM server, which updates the TBSM graphical user interface (GUI).

Status changes are captured by RODM and passed to an NT-based set of services, which records them as events on the appropriate resources.

The Mapping of State Changes to TBSM Events

TBSM monitors changes to the RODM *states* field for objects that are registered for monitoring. In order for changes to the *states* field to participate in the TBSM propagation model, they are mapped to TBSM messages and exceptions.

TBSM monitors the RODM *DisplayStatus* field. Changes to the *DisplayStatus* field are mapped to messages, which reflect the current state of the object.

Refer to Appendix C on page 49 for the field changes mapped to TBSM events.

Architecture

The following figure and event flow explains the processing data flow of events in a numbered format as they move through the various architectures of TBSM.

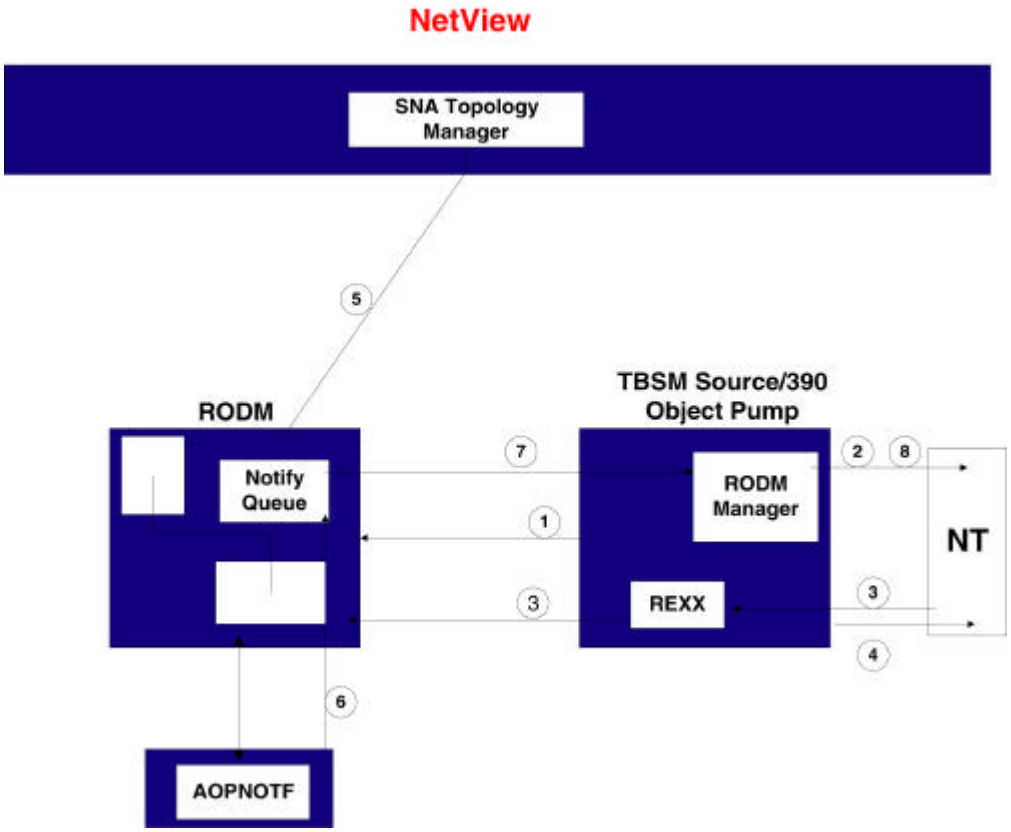


Figure 3. SNA Events from RODM to TBSM Flow

Event Flow

1. RODM Manager connects to RODM and identifies the notify method, AOPNOTF.

2. RODM Manager sends a RODM available message to NT.
3. NT sends registration requests, which cause the TBSM Source/390 Object Pump to connect the notify method to the objects to be monitored in RODM.
4. A message containing the current field data from RODM is returned to NT for all monitored objects.
5. SNA Topology Manager causes a change to the data in the field of a monitored object
6. A change to the monitored field data of a monitored object causes the notify method to send a message to the notify queue.
7. The RODM Manager retrieves the message from the notify queue.
8. A message containing the new data from the object field is sent to NT.

5

User Interface

This chapter describes the GUI support for SNA/APPN objects.

It contains the following interface elements and functions:

- RODM Property Sheet and Tab
- SNA Network Property Sheet

RODM Property Sheet and Tab

Network components are constantly being monitored and from time to time within RODM their state changes. When this happens to a registered object, TBSM is appraised of the change through specially developed notification methods that are associated with the object. These methods, which are keyed to the RODMObjectClass fields, ensure that the state of the object in RODM reflects in TBSM.

After the object receives an alert icon (Red or Yellow) notifying you that a problem exists, you double-click the object to view the TBSM Attributes tab on the Property Sheet; you then examine the cause of the alert. The Current Events grouping, located on the lower portion of the Property Sheet, presents a list view of messages and exceptions. Exceptions appear at the top of the list. The current events appear in the white area, while the historical events appear in the shaded area.

TBSM provides a RODM tab on the Property Sheet for all RODM objects you monitor.

You click the RODM tab to view an entire dialog dedicated to the RODM object.

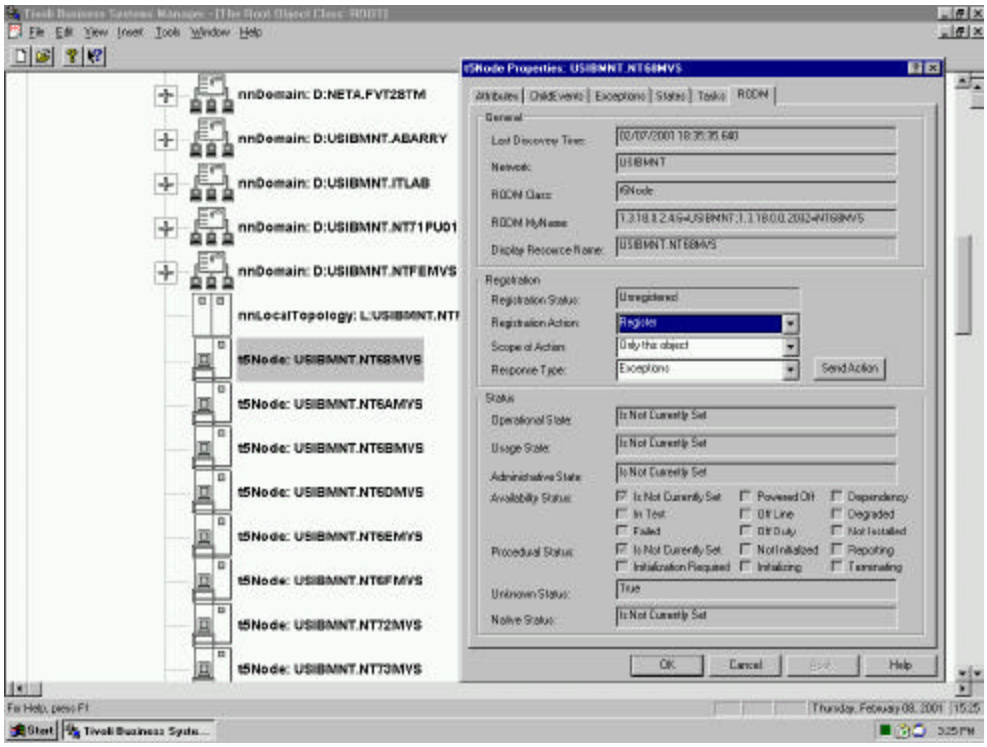


Figure 4. RODM tab dialog

The RODM tab dialog consists of the following major groupings:

- General
- Registration
- Status

General Fields

The General grouping contains the following Read-Only fields:

- Last Discovery Time - Lists the actual date and time of last RODM discovery or last RODM event.
- Network - Name of the SNA/APPN network ID
- RODM Class - RODM class name of the object.
- RODM MyName - RODM *MyName* field that distinctly indentifies the object (instance name). This also referred to as the relative distinguished name (RDN).
- Display Resource Name - RODM *DisplayResourceName* is important for identifying the object when communicating with the Network Management Console (NMC) application.

Registration Fields

The Registration grouping consists of the following fields:

- Registration Status (Read-Only) - States if the RODM object is being monitored by TBSM. If it is registered, events are received; if unregistered, no events are received directly from RODM.

  Possible values for this field are the following:

  - Registered
  - Unregistered

- Registration Action - This field allows you to select a TBSM registration operation to perform on the RODM object. The result of this action establishes or removes the notice for receiving events from RODM.

  The possible values for this field are the following:

  - Register - If an object is registered, RODM specific events are placed on this object.
  - Unregistered - If an object is unregistered, no RODM specific events are placed on this object.
  - Reregister - Retransmits the Registration Status to the OS/390 ensuring current registration selections.

- Scope of Action - This field allows you to select which objects receive the Registration Action.

Possible values for this field are the following:

- Only this object
- Only children of this object - Refers to the children of this object as they reside in the BUSC, ***not*** including the current selected object.
- This object and its children - Refers to the children of this object as they reside in the BUSC, including the current selected object.

■ Response Type - This field describes the level of feedback that is returned to the TBSM system on the use of the registration action. This feedback is in the form of events that are posted directly to the objects. You view these events on the Attributes tab of the Property Sheet, under the Current Events grouping.

For further information about possible events, refer to the following publication:

NetView V2.4 RODM Programming Guide.

Document number SC31-7095

The possible values for this field are the following:

- Definite - A confirmation event is posted to the object regardless of the success of the registration.
- Exceptions - An exception event is posted to the object only in the case of an unsuccessful registration.
- Off - No events are posted to the object due to registration.

Object Registration

The final three fields of the Registration grouping are used with the Send Action button to transmit registration notices through TBSM to the OS/390 system.To perform a registration action, select the appropriate action, scope, and response type from the fields, and click the Send Action button.

A simple registration action is the registration of a single object with a guaranteed event response.

To perform this action, you select the following:

- Registration Action: **Register**
- Scope of Action: **Only this object**
- Response Type: **Definite**

If you examine the Registration Status field for a single object and its children, and find they are configured properly under TBSM, but you want to reregister the objects with the OS/390 system to be certain they are being monitored, you invoke the following options. You invoke the following options, which are enacted on a single object, reregister the object, its children, and only produce registration error events if they occur:

- Registration Action: **Reregister**
- Scope of Action: **This object and its children**
- Response Type: **Exceptions**

To unregister the children of a single object and receive no events, you select the following options:

- Registration Action: **Unregister**
- Scope of Action: **Only children of this object**
- Response Type: **Off**

  Note: This operation does not produce any feedback events, even in the case where the unregistration of an object produces an error. You are not aware of the success of the operation, nor able to check the results of the operation. You should use the response type of **Off** with extreme caution.

Status Fields

This grouping is present when a particular RODM object maintains status information. The Status fields exhibit the open systems interconnection (OSI) and native status of the object.

The Status fields, which are Read-Only, contain the following:

- Operational Status - Indicates the operability of managed object or resource.
- Usage State - Indicates the current usage of the managed object or resource.
- Administrative State -Indicates whether the managed object or resource is administratively prohibited from being used.
- Availability Status - Indicates the availability of the managed object or resource. The specific check box is marked indicating that the condition exists.
- Procedural Status - Indicates the current initialization status of the managed object, or resource.The specific check box is marked indicating that the condition exits.
- Unknown Status - Indicates whether the state of the managed object or resource is unknown.
- Native Status - Indicates product-specific status as a supplement to OSI status attributes.

The following table lists the corresponding values for each status fields:

Status Field	Value Description
Operational State	■ Disabled ■ Enabled ■ Is Not Currently Set
Usage State	■ Idle ■ Active ■ Busy ■ Is Not Currently Set
Administrative State	■ Locked ■ Unlocked ■ Shutting Down ■ Is Not Currently Set

Status Field	Value Description
Availability Status	■ Is Not Currently Set ■ In Test ■ Failed ■ Power Off ■ Off Line ■ Off Duty ■ Dependency ■ Degraded ■ Not Installed
Procedural Status	■ Is Not Currently Set ■ Initialization Required ■ Not Initialized ■ Initializing ■ Reporting ■ Terminating
Unknown Status	■ False ■ True

Status Field	Value Description
Native Status	■ Active ■ Active With Session ■ Inactive ■ Never Active ■ Pending Active ■ Pending Inactive ■ Connectable ■ Routable ■ Operative ■ Congested ■ Released ■ Reset ■ Inoperative ■ Pending Active Or Inactive ■ Released Or Reset ■ Force Native Status To FF ■ Is Not Currently Set

Depending on the RODM object you select from your view, the Status fields do not appear on the lower portion of the dialog.

The following RODM objects do not have a status:

- enLocalTopology
- IDCircuit
- LUGroup
- nnDomainNetworkCluster
- nnDomain
- nnDomainNetwork
- nnLocalTopology
- subareaTGCircuit

All other RODM objects appear with Status fields. Refer to Appendix A on page 35 for a complete list of monitored objects.

Identifying the RODM Operating System

When you define a SNA/APPN Network to TBSM, you must also define which Operating System to use in communicating with the RODM repository, which contains the objects that are in the network. Doing so establishes the communication path between TBSM and RODM.

The following SNA/APPN Network Property Sheet contains a drop-down box, Monitoring Operating System, within the Additional grouping, which you use to select the appropriate Operating System.

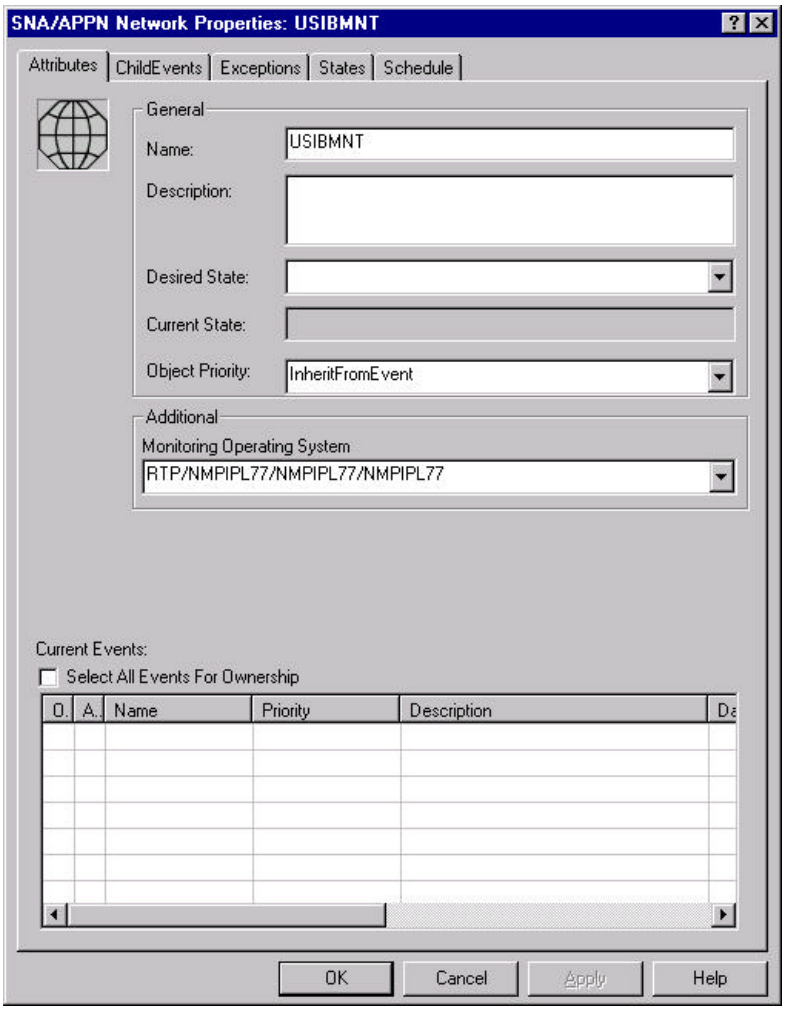


Figure 5. SNA/APPN Network Property Sheet

Depending on where you place the Network object in the TBSM hierarchy, the list of available Operating Systems are appropriately limited in scope (you can place a SNA/APPN Network under any Enterprise, Complex, Machine, LPAR, or Operating System in the TBSM hierarchy). If the you place a Network object under an

Enterprise, then you can select any Operating System in the Enterprise. Consequently, if you place the Network object under an Operating System, you can only select that Operating System.

Note: It is important to recognize that you **must** select a valid Operating System for each SNA/APPN Network defined to TBSM before monitoring of that network can proceed.

Chapter 6. Configuring RODM/TBSM Interface
Chapter 6. Configuring RODM/TBSM Interface

6

Configuring RODM/TBSM Interface

This chapter explains the necessary steps for discovery and enabling event flow from RODM to TBSM.

Enabling TBSM OS/390 Components

To enable TBSM OS/390 components:

1. Install the data sets through the System Modification Program Extended (SMP/E) for TBSM maintenance.
2. Prepare the RODM unload and GTMAOPE0 JCL. Refer to the sample in the SGTMSAMP library. This job should run after all configuration is completed.
3. You need to add the following parameters to the TBSM Source/390 Object Pump Startup parameters to activate the RODM interface. For the entire contents of the Startup parameters, refer to the section "TBSM Source/390 Object Pump Startup Parameters" in the *Tivoli Business Systems Manager Installation and Configuration Guide.*

RODM_NAME=*name* specifies RODM 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 omitted, then network status data is not 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 then the TBSM Source/390 Object Pump job name is used as the RODM user name. If a system authorization function (SAF) is providing security to RODM then you code RODM_USERID=' 'to force a blank userid to be used. The userid used must have at least RODM level 5 security.

Note: Refer to the *NetView for OS/390 Administration Guide* for further details.

RODM_NOTIFY_METHOD=AOPNOTF | *Method name* specifies the name of the RODM method that is 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. If so, you must relink with the new name to the supplied RODM method in order to match the new name that you specify.

RODM_NOTIFY_QNAME=AOPNOTFQ | *Notify queue name* specifies the name of the RODM notify queue that is 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.

Note: In addition, the RODM method **AOPNOTF** needs to be made available to the RODM address space in the STEPLIB concatenation or LNKLIST definitions. **AOPNOTF** is available via a *NetView PTF UW76035 for Version 1.2 and UW76034 for 1.3.*

Configuring the TBSM Server

The following section describes the configuration of the NT-based components for RODM discovery.

TBSM Object Configuration

You place a SNA/APPN Network object under any Enterprise, Complex, Machine, LPAR, or Operating System in the TBSM hierarchy. You select the appropriate high-level object (Enterprise,

Complex....) and right-click, select Insert ûSNA/APPN Network.The name of this object ***must*** be exactly the name of the Network ID as it appears in the VTAM definition.

Note: You must select a valid Operating System for each SNA/APPN Network defined.

IP Listener Configuration

The ASIMVSIPListener is used to receive files sent by the GTMAOPE0 utility. The ASIMVSIPListenerSvc must run on the SQL Server.

For installations prior to TBSM Version 1.5, you must manually invoke the following:

1. Copy the *ASIMVSIPListenerSvc.exe* from *Tivoli Manager\bin* on the Event Handler machine to the same location on the SQL Server machine.
2. Install SNA client software on the SQL machine. Refer to the *TBSM Installation and Configuration Guide* for more details.
3. From a command prompt on the SQL Server issue:

   ```
   ASIMVSIPLISTENERSVC -SERVICE
   ```

4. Make any Registry modifications as specified in the following section and start the service.

Registry Modifications for the IP Listener

On hosts running the ASIMVSIPListenerSvc, you configure the appropriate entry in the NT Registry for the processing to be completed. The following Registry keys contain parameters that are required for the RODM Discovery process.The values displayed are set by default and should be modified for your installation.

This setting defines the port address to communicate with GTMAOPE0. The default is 1021. You only need to code this if your JCL specifies a port other than 1021.

HKEY_LOCAL_MACHINE \ SOFTWARE \ Accessible Software,Inc. \Access1 \ 1.0 \ Components \ ASIMVSIPListenerSvc \ Settings Port 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.

HKEY_LOCAL_MACHINE \ SOFTWARE \ Accessible Software,Inc. \Access1 \ 1.0 \ Components \ ASIMVSIPListenerSvc \ Settings \ CommandAliases RODMDISCOVERY sh CreateDiscoveryBatch.ksh –F5 –A0 %s

This setting defines the IP clients that are authorized to run GTMAOPE0. This is specified on the TCPIP_ADDRESS or TCPIP_NAME control card of GTMAOPE0.

HKEY_LOCAL_MACHINE \ SOFTWARE \ Accessible Software,Inc. \Access1 \ 1.0 \ Components \ ASIMVSIPListenerSvc \ Settings \ValidClients

SQL Server Job Configuration

The *CreateDiscoveryBatch.ksh* script defined in the Registry moves and renames the downloaded RODM file and assigns it a state of ENQUEUED in the *DiscoveryBatch* table. This occurs each time the GTMAOPE0 utility is used to download RODM files to the SQL Server.

If a system variable TEMP is not defined to the NT system where this script runs, the files will be moved to the WINNT directory. We recommend that you define a system variable TEMP.

Note: The TBSM administrator uses the SQL Enterprise Manager to define a run schedule for each of the Structured Query Language (SQL) Server jobs. The schedule determines how often the job runs and in what time window.

The two SQL Server jobs are the following:

- *RODM Discovery Load*
- *RODM Discovery Process*

RODM Discovery Load job takes any RODM files in the SQL *DiscoveryBatch* table that are in the ENQUEUED State and loads the associated file into the SQL Server database. If an error occurs, the State is set to LOAD_ERROR; otherwise the State is set to LOADED. In either case an appropriate message is logged.

Chapter 6. Configuring RODM/TBSM Interface
Chapter 6. Configuring RODM/TBSM Interface

RODM Discovery Process job performs the processing required on any RODM files in the SQL table *DiscoveryBatch* that are in the LOADED state. The job updates TBSM object hierarchy. When a *Discovery Batch* is processed, it can place a significant load on the database, therefore, we recommend you schedule the job for low-activity periods. When the *RODM Discovery Process* ends successfully, the state is set to completed.

State can be viewed by SQL; *Select batchid,pstat,ctime from DiscoveryBatch* and observe the pstat.

You find the following definitions in the SQL tabe.*DiscoveryBatchPStatVal.*

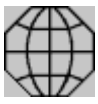
Pstat	Name
0	ENQUEUED
1	LOADED
2	INPROGRESS
3	COMPLETED
4	ALLOCATED
-1	LOAD_ERROR
-2	PROCESS_ERROR
-3	(Undefined)
-4	ALLOCATE_ERROR

Appendix A. SNA/APPN Class Descriptions
Appendix A. SNA/APPN Class Descriptions

A

SNA/APPN Class Descriptions

The following table provides the icon (if available), class name, and detailed description of each network object monitored by TBSM.

Icon	Class Name	Description
	nnDomain	Provides aggregation for the *nnDomainNetwork member objects, and the interdomainNetworkCircuit memberArcs objects that interconnect the*nnDomainNetwork objects.
	nnDomainNetwork	Provides aggregation for: Advanced Peer-to-Peer Networking (APPN) network node of the *nnDomain object as an NN server node; APPN branch node of the *nnDomain object; APPN end nodes the network node serves; APPN transmission group circuits that support CP-CP sessions.
	nnDomainNetwork Cluster	Provides aggregation for the *nnDomain member objects, and the *interdomainCircuit memberArcs objects that interconnect the *nnDomain objects.

Icon	Class Name	Description
	appnBrNN	To nodes in the WAN including its network node server appears to be an APPN end node. To end nodes in its branch (That is, end nodes to which it is connected by a transmission group (TG) it defines as a branch downlink.), it appears to be a network node.
	appnEN	SNA type 2.1 node that supports its own end users, providing limited directory and route selection services, as well as management services.
	appnNN	SNA type 2.1 node that provides the following services to end nodes: directory, management, route selection, performs intermediate routing of data on sessions that traverse it.
	appnTG	Provides a route for a session stage between two logically adjacent nodes. Uniquely identified by transmission group number. A route is composed of a series of transmission group numbers and names of associated adjacent nodes.
	appnTGCircuit	Represents an appnTransmissionGroupCircuit that is connected to a node with the following NTRI-like resources: NTRI, FrameRelay, ESCON, and Ethernet.
	appnTGCircuitCN	Comprises one or two underlying appnTransmissionGroup objects, and provides a composite status of the appnTransmissionGroup objects.
	ntriTypeAppnTgCircuit	Represents an APPN connection to a composite node (interchange node) or a t5Node that traverses the subarea part of the network

Icon	Class Name	Description
	IDCircuit	Defines the fields that apply to aggregate objects representing APPN transmission group (TG) circuits between two APPN network nodes or between a network node and a virtual routing node.
	IDNetworkCircuit	Defines the fields that apply to aggregate objects representing APPN intersubnetwork TG circuits.
	xDomainResource	Logical unit (for example, an application program, a peripheral node, or a terminal) that is controlled by another VTAM domain.
	xDomainResource Manager	Part of a system services control point (SSCP) that supports cross-domain session (SSCP-SSCP or CDRM-CDRM session) setup and take down.
	definitionGroup	Defines the fields that apply to major node objects reported by the VTAM agent. A definitionGroup is a VTAM major node, which is a grouping of similar resource definitions.
	interchangeNode	SNA node that has both the type-5 node function of a subarea node, including an SSCP, and the type 2.1 node function of an APPN network node. The interchange node provides the capability to interconnect an SNA subarea network and an APPN network.
	lenNode	Low-Entry Networking. SNA type 2.1 node that: Supports its users and provides the following services: link-connection, local directory and management.

Icon	Class Name	Description
	logicalLink	Provides a connection between two logically adjacent nodes.
	logicalUnit	Access point for applications, transaction programs, and end-users to the network.
	luGroup	Set of logical units (logical unit (LU))s that have been grouped together under one generic resource name. The generic resource function is only available in a sysplex environment running under a release of OS/390 that supports coupling facility services.
	migrationDataHost	SNA node that has both the type-5 node function of a subarea node, including an SSCP, and the type 2.1 node function of an APPN end node.
	DLCPort	Provides a data link control (DLC) connection point within a node for a logical link
	enLocalTopology	Used for creating enLocalTopology views. An object of *enLocalTopology represents a model for an APPN end node (or migrationDataHost), its logically adjacent nodes (if any), and the transmission groups (if any) between the local EN and the adjacent nodes.

Icon	Class Name	Description
	nnLocalTopology	Represents a model for an APPN network node (or interchangeNode), its logically adjacent nodes (if any), and the transmission groups (if any) between the local NN and the adjacent nodes. The object is an aggregate object, but has no aggregation links.
	snaNode	Used when the class of a node object is unknown. This class is used when the SNA topology manager is monitoring an individual logicalUnit or lu Collection and the owning node name is known, but the class of the owning node is not known.
	subareaTGCircuit	Represents the subarea connection between two subarea nodes.
	t2_1Node	Represents an adjacent node defined to the reporting agent node, where: the adjacent node type is not yet known to the agent node or the topology manager; or a logical link between the agent node and the adjacent node has not been activated.
	t4Node	Used to interconnect subarea networks. A gateway node translates between the network name and address used by one subarea network to represent a resource, and the network name and address used by another subarea network to represent the same resource.
	t4NodeGateway	SNA type-4 node provides communication controller support for single-domain, multiple-domain, and interconnected network capability. It can also provide gateway function, intermediate routing function, and boundary function

Icon	Class Name	Description
	t5Node	Represents a type-5 SNA node. A type-5 node is a subarea node containing a system services control point (SSCP) that has hierarchical control of type 4 nodes and peripheral nodes. A type-5 node is usually referred to as a host node.
	virtualRoutingNode	Connection network used in an APPN network, not a node. Used by APPN nodes to calculate and establish direct routes for sessions between origin and destination nodes within a shared-access transport facility, such as a token-ring local area network (LAN).

B

Messages and Exceptions

The following tables display the RODM messages and exceptions for the TBSM interface.

RODM Exceptions for SNA Objects

RODM Status Field	Exception	Alert State	Priority
administrativeState	Administrative State is not currently set	Yellow	High
	Administrative State locked	Red	High
	Administrative State shutting down	Red	High
	Administrative State unlocked	Green	Ignore
availabilityStatus	Availability Status degraded	Red	High
	Availability Status dependency	Red	High
	Availability Status failed	Red	High

RODM Status Field	Exception	Alert State	Priority
	Availability Status in test	Red	High
	Availability Status not installed	Yellow	High
	Availability Status off duty	Red	High
	Availability Status offline	Red	High
	Availability Status power off	Red	High
nativeStatus	NativeStatusactive	Green	Ignore
	Native Status active with session	Green	Ignore
	Native Status congested	Yellow	Ignore
	Native Status connectable	Yellow	Ignore
	Native Status inactive	Yellow	Ignore
	Native Status inoperative	Yellow	Ignore
	Native Status is not currently set	Yellow	Ignore
	Native Status never active	Yellow	Ignore
	NativeStatusoperative	Yellow	Ignore
	Native Status pending active	Yellow	Ignore
	Native Status pending active or inactive	Yellow	Ignore

RODM Status Field	Exception	Alert State	Priority
	Native Status pending inactive	Yellow	Ignore
	Native Status released	Yellow	Ignore
	Native Status released or reset Native	Yellow	Ignore
	Status reset	Yellow	Ignore
	NativeStatus ratable	Yellow	Ignore
operationalState	Operational State disabled	Red	Critical
	Operational State is not currently set	Yellow	High
proceduralStatus	Procedural Status initialization required	Yellow	High
	Procedural Status initializing	Yellow	High
	Procedural Status not initialized	Yellow	High
	Procedural Status reporting	Yellow	High
	Procedural Status terminating	Yellow	High
unknownStatus	Unknown Status	Red	High
usageState	Usage State active	Green	Ignore
	Usage State busy	Green	Ignore
	Usage State idle	Green	Ignore
	Usage State is not currently set	Green	Ignore

Registration Status Messages for SNA Objects

Exception	AlertState	Priority
RODM REGISTER FAILED	Yellow	High
RODM REGISTER SUCCEEDED	Green	High
RODM UNREGISTER FAILED	Yellow	High
RODM UNREGISTER SUCCEEDED	Green	High

RODM Available Messages for SNA Objects

Message	AlertState	Priority
Display Status degraded (133)	Red	Medium
Display Status intermediate (131)	Yellow	Ignore
Display Status low satisfactory (145)	Green	Low
Display Status low unsatisfactory (161)	Yellow	Low
Display Status medium satisfactory (144)	Green	Medium
Display Status medium unsatisfactory (160)	Yellow	Medium

Message	AlertState	Priority
Display Status negative user-defined (152)	Yellow	Ignore
Display Status negative user-defined (153)	Yellow	Ignore
Display Status negative user-defined (154)	Yellow	Ignore
Display Status negative user-defined (155)	Yellow	Ignore
Display Status negative user-defined (156)	Yellow	Ignore
Display Status negative user-defined (157)	Yellow	Ignore
Display Status negative user-defined (158)	Yellow	Ignore
Display Status negative user-defined (159)	Yellow	Ignore
Display Status positive user-defined (136)	Yellow	Ignore
Display Status positive user-defined (137)	Yellow	Ignore
Display Status positive user-defined (138)	Yellow	Ignore
Display Status positive user-defined (139)	Yellow	Ignore

Message	AlertState	Priority
Display Status positive user-defined (140)	Yellow	Ignore
Display Status positive user-defined (141)	Yellow	Ignore
Display Status positive user-defined (142)	Yellow	Ignore
Display Status positive user-defined (143)	Yellow	Ignore
Display Status satisfactory (129)	Green	High
Display Status severely degraded (134)	Red	High
Display Status unknown (132)	Yellow	Ignore
Display Status unsatisfactory (130)	Yellow	High

RODM-Related Exceptions and Messages for an Operating System

Exception	AlertState	Priority
RODM STATUS DOWN	Red	High
RODM STATUS UP	Green	High

Message	AlertState	Priority
ASI_05\01 (RODM Connection Status)	Unknown	Ignore

Message	AlertState	Priority
ASI_05\02 (RODM Object registration)	Unknown	Ignore

Appendix B. Messages and Exceptions
Appendix B. Messages and Exceptions

C

GTMAOPE0 UTILITY

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 ASIMVSIPListenerSvc.

GTMAOPE0 Control Cards

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 ASIMVSIPListenerSvc. 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.

Note: Refer to the TCP/IP manuals for more information regarding the TCP/IP parameters. Refer to the *TBSM Distributed Installation and Configuration Guide* to determine the IP address or hostname to use.

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.

Note: Refer to the *TBSM Distributed Installation and Configuration Guide* to determine the Port number to use.

TCPIP_JOBNAME=tcpip_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 RODM data received, you may need to increase the TIMEOUT value.

CODEPAGE=codepage_number | 037

Specifies the codepage number to the ASIMVSIPListenerSvc 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.

Note: Refer to the *TBSM Distributed Installation and Configuration Guide* to determine the CODEPAGE to use.

TEXT=NO | YES

Appendix C. GTMAOPE0 UTILITY
Appendix C. GTMAOPE0 UTILITY

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.

Note: Refer to the *TBSM Distributed Installation and Configuration Guide* to determine the proper TEXT mode of operation.

COMMAND=command_name

Specifies a command name sent to the ASIMVSIPListenerSvc 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.

Note: Refer to the *TBSM Distributed Installation and Configuration Guide* to determine the command name to use.

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.

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 ASIMVSIPLIstenerSvc service is to process. This value is always 2 bytes less than the PHYSICAL BUFFER value.

The ASIMVSListenerSvc

The TRACE and LOG DDNAMES are used in conjunction with the ASIMVSIPListenerSvc 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:

- Output that is generated by the GTMAOPE0 utility
- Output that is generated by the GTMAOPE0 utility
- The LOG file generated by the ASIMVSIPListenerSvc running on Windows/NT

File Number: