



Tivoli Business Systems Manager
System Automation for OS/390 Release Notes
Version 1.5



Tivoli Business Systems Manager
System Automation for OS/390 Release Notes
Version 1.5

Tivoli Business Systems Manager System Automation for OS/390 Release Notes

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, NetView, OS/2, RS/6000, Tivoli Enterprise, Tivoli Enterprise Console, 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, Windows 2000, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

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

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

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	v
Accessing Publications Online	v
Ordering Publications	v
Providing Feedback about Publications	v
Contacting Customer Support	vi
Release Notes.....	1
Overview	1
External Data Interface	1
External Data Interface Data Processing in Tivoli Business Systems Manager.....	2
Environmental Prerequisites.....	2
Functional Components.....	2
Required Application Capture Modifications	3
Scenario Event Flow	5
Integration with Tivoli Business Systems Manager.....	5
Status Change	6
Error Handling	7
Automation (Starting/Ending)	7
Architecture	9
Event Flow	9
Object Discovery.....	10
Processing Elements	10
Data	10
Data Samples	10
Automation Control File	10
File Header Information.....	10
Operating System (Image) Data	11
Member Links.....	12
Member Detail	12
Discovery and Extraction Methodology	12
Step 1: SYSNAME Pointer Capture.....	12
Step 2: Member Links Capture	13
Step 3: Member Detail Capture	14
Step 4: Tivoli Business Systems Manager Loader File Creation	15
Appendix A. Status Messages for TBSM Objects	17
Appendix B. Messages for Registered Object Types.....	27

Batch	27
CICS.....	28
CMAS	28
DB2 PM	29
DB2 Subsystem.....	30
IDMS	30
IMS	31
Omegamon	32
Started Task	33
Appendix C. Object Discovery Process Diagrams	35
Data Flow Diagram.....	36
Bulk Discovery Data Flow Using the SA OS/390 Automation Control File.....	36
Appendix D. List of Files Used in Object Discovery	37
Automation Control File (ACF).....	37
Image Header File.....	37
Anchor Member File	37
Member Detail File	37
Appendix E. Configuration of Event-Based Discovery	39
SA390ResourceTypeMap Table	39
Supported Resource Type Class Ids	40
Configuring the SA390ResourceTypeMap Table.....	40
Configuring State Change Message Format	41

Preface

Tivoli Business Systems Manager (TBSM) provides a robust, system management functionality. In support of the operational perspectives of that functionality, TBSM installs components on both the OS/390 and NT platforms.

Accessing Publications Online

The following sections describe how to access publications online, order publications, provide feedback about publications and contact customer support.

The Tivoli Customer Support Web site (<http://www.tivoli.com/support/>) offers a guide to support services (the *Customer Support Handbook*); frequently asked questions (FAQs); and technical information, including release notes, user's guides, redbooks, and white papers. You can access Tivoli publications online at <http://www.tivoli.com/support/documents/>. The documentation for some products is available in PDF and HTML formats. Translated documents are also available for some products.

To access most of the documentation, you need an ID and a password. To obtain an ID for use on the support Web site, go to <http://www.tivoli.com/support/getting/>.

Resellers should refer to <http://www.tivoli.com/support/smb/index.html> for more information about obtaining Tivoli technical documentation and support.

Business Partners should refer to "Ordering Publications" for more information about obtaining Tivoli technical documentation.

Attention: The following note is an example of exceptional information. If your documentation requires similar, exceptional information, add it in the appropriate section (however, it is likely that your documentation does not require any additional notes or addenda). In all instances, remove this Attention element.

Note: For NetView OS/390 customers, additional support is also available on the NETVIEW CFORUM (Customer Forum) through the IBMLink system. This forum is monitored by NetView developers who answer questions and provide guidance. When a problem with the code is found, you are asked to open an official problem management record (PMR) to get resolution.

Ordering Publications

Order Tivoli publications online at http://www.tivoli.com/support/Prodman/html/pub_order.html or by calling one of the following telephone numbers:

- U.S. customers: (800) 879-2755
- Canadian customers: (800) 426-4968

Providing Feedback about Publications

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

- Send e-mail to pubs@tivoli.com.

-
- Fill out our customer feedback survey at <http://www.tivoli.com/support/survey/>.

Contacting Customer Support

If you need support for this or any Tivoli product, contact Tivoli Customer Support in one of the following ways:

- Submit a problem management record (PMR) electronically from our Web site at <http://www.tivoli.com/support/reporting/>. For information about obtaining support through the Tivoli Customer Support Web site, go to <http://www.tivoli.com/support/getting/>.
- Submit a PMR electronically through the IBMLink™ system. For information about IBMLink registration and access, refer to the IBM Web page at <http://www.ibm.link.ibm.com>.
- Send e-mail to support@tivoli.com.
- Customers in the U.S. can call **1-800-TIVOLI8 (1-800-848-6548)**.
- Customers outside the U.S. should refer to the Tivoli Customer Support Web site at <http://www.tivoli.com/support/locations.html> for customer support telephone numbers.

When you contact Tivoli Customer Support, be prepared to provide the customer number for your company so that support personnel can assist you more readily.

1

Release Notes

Overview

The old adage of one product for one task has been replaced in today's enterprise computing environment by a solution that provides you with high-end automation capabilities, integrated systems, and network management operations. If consolidation and integration are important strategies for you, then you need tools built on a common base that extends from network management to systems management. If automation is essential, then you need tools that make it as easy as possible for operations to implement automation. The broad scope and power of *System Automation for OS/390* (SA OS/390), a Tivoli NetView for OS/390-based application, enables you to cover monitoring, automation, and manual control of the full operational life cycle of your enterprise-wide systems hardware and software resources.

The overall functionality of Tivoli Business Systems Manager has been enhanced by interfacing with automation based applications. SA OS/390 exploits Tivoli Business Systems Manager's ability to present graphic images of systems, subsystems, and applications of an enterprise in order to detect the status of automation routines as well as critical SA OS/390 events.

In addition to real-time monitoring of resources for alert conditions in the MVS part of the OS/390 environment, this interface also notifies customers of any automation in progress for a failed resource. For example, if Tivoli Business Systems Manager reports a resource in a failed status (indicated by a Red alert), an automation routine is used to resolve the failed status. The automation routine indicates to Tivoli Business Systems Manager, via the *External Data Interface* (EDI), that automation is attempting a corrective action for this object (resource) by posting an "Automation in Progress" indicator on the alert icon similar to the Taking Ownership principle. Once the automation routine has resolved the critical situation, it notifies Tivoli Business Systems Manager (via the EDI) that the automation function has successfully completed. Tivoli Business Systems Manager then returns the object in question to "available" (by removing the Red alert). However, if the automation fails the icon is removed and replaced with an "Exception" icon.

Tivoli Business Systems Manager notification by SA OS/390 enables you to monitor and manage system-specific automation events directly from the same console as other events in your OS/390 environment. The following sections describe the components of the interface of Tivoli Business Systems Manager with SA OS/390.

External Data Interface

External Data Interface (EDI) provides the vehicle whereby text-based messages or exceptions, which reside upon the same MVS Operating Systems, uses cross-memory services to pass messages or exceptions to Tivoli Business Systems Manager. EDI inserts the

time-stamp and sequence number to the message or exception, adjusts the final length, and sends the data to Tivoli Business Systems Manager under NT.

It is important to note that these messages and exceptions are passed *directly* to the *TBSM Source/390 Object Pump* (by EDI) and are not displayed upon the system console.

External Data Interface Data Processing in Tivoli Business Systems Manager

Tivoli Business Systems Manager uses the EDI to allow external products (SA OS/390) to pass data directly to the TBSM Source/390 Object Pump and on to NT without the need for major code changes or additions in order to handle every new message type from each product.

The external products themselves build the bulk of the messages to be sent to NT in the message API format already defined along with the action types. Therefore, the product interfaces, which capture the data and send it to the TBSM Source/390 Object Pump/NT via EDI, define the format of the records.

Each product interface prefixes these messages with a unique 8-byte message number that the TBSM Source/390 Object Pump initializes via a trap. The TBSM Source/390 Object Pump Message Handler inserts the time-stamp and sequence number into the record, adjust the final length, and send the data, as is, to NT.

The traps for the messages are pre-defined within the TBSM Source/390 Object Pump (Object Pump) startup parameters data, therefore, no registration data is required to activate these traps. Default parameters are supplied in the Object Pump startup parameters module enabling you to add the minimum details to activate the traps.

Environmental Prerequisites

In order to support the processing necessary to facilitate the passing of data from SA OS/390 to Tivoli Business Systems Manager through EDI, some environmental settings must be established.

System Automation for OS/390:

- Edit the SA OS/390 procedure, commonly called (**INGESO**) and add these statements:
 - **SYS5.TM390.SGTMEEXEC** to the **DSICLD DD** concatenation
 - **SYS5.TM390.SGTMMODS** to the **STEPLIB DD** concatenation

Functional Components

These following functional components facilitate the ability to pass data from SA OS/390 through EDI to Tivoli Business Systems Manager.

Application Capture

The Application Capture with a minor modification within the SA OS/390 application enables the capture and formatting of the message and exception information generated by SA OS/390. Once captured, this information is passed to the REXX program for further action.

REXX Program

This compiled program, *GTMEDISA*, resides within the Tivoli Business Systems Manager REXX library. This program formats the passed message or exception, providing additional

parameters and information to the message/exception as required for use in the SA OS/390-EDI-to-Tivoli Business Systems Manager interaction. Upon completion, the REXX program invokes the EDI for further action.

Note: Your source code needs to change from "I" to "!" when using X'4F'. This varies from country to country.

EDI

This program and related functionality is provided with Tivoli Business Systems Manager. The program, *GTMRXEDI* resides in the load module library. This program adds date and time information, as well as assignment of a sequence number, and provides the functionality to pass the formatted message or exception information to the TBSM Source/390 Object Pump (Object Pump).

TBSM Source/390 Object Pump

Once the message or exception information is sent to the TBSM Source/390 Object Pump through EDI, the TBSM Source/390 Object Pump traps the messages or exception, and provides the respective information to Tivoli Business Systems Manager for appropriate action.

Required Application Capture Modifications

To support passing of data from a source product or application to Tivoli Business Systems Manager 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 SA390's REXX EDI:

```
'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 STCTEST1/* OBJECT NAME */
'GTMEDISA' TOKEN 08 ABENDING/* STATE CHANGE */
'GTMEDISA' TOEKN 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.

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 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 contains Data Type and Data. In the example, the Data Type is 05 (or SMFID) and 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 was passed, the record being passed to the PUMP would be printed out. This record is useful for diagnosing a problem with the EDI.

Notes:

When passing text (for example, 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:

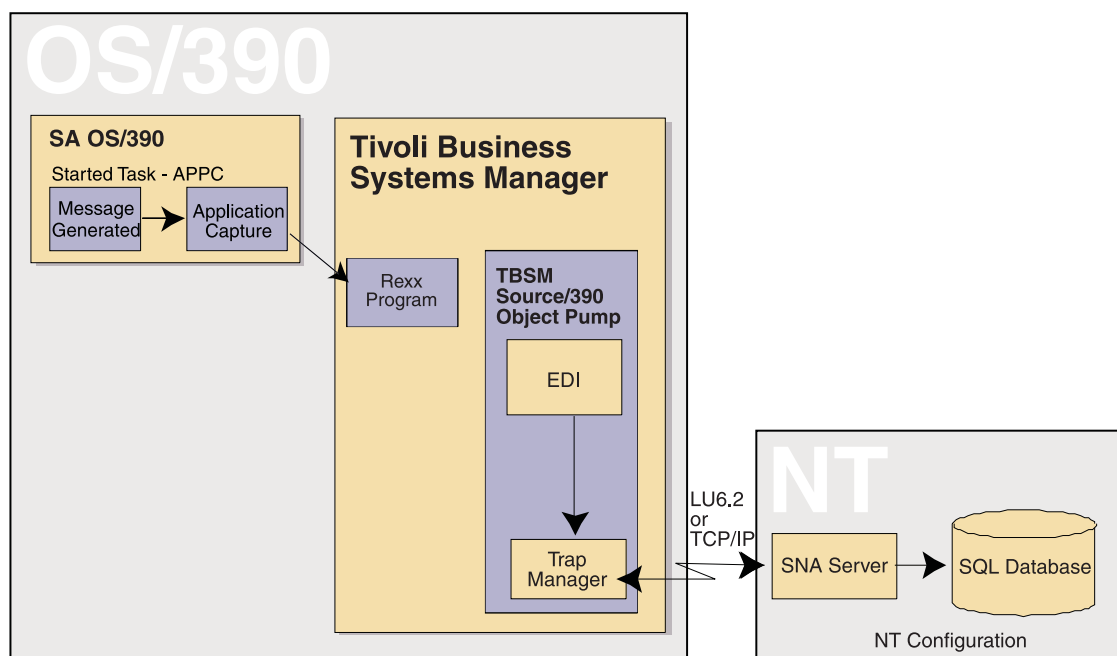
```
'GTMEDISA' START_OF_DATA
Token = "GTM" || rc
'GTMEDISA' TOKEN 07 02/* SA390 MESSAGE EVENT */
'GTMEDISA' TOKEN 05 IP01/* SMF ID */
'GTMEDISA' TOKEN 03 BATCH001/* OBJECT NAME */
'GTMEDISA' TOKEN 08 SAM001I/* MESSAGE ID */
'GTMEDISA' TOKEN 49 "Test message from SA390" /* MESSAGE TEXT */
'GTMEDISA' TOKEN END_OF_DATA TRACEOFF
```

Additional examples of using the REXX EDI can be found in section "External Data Interface and Automation Products" of the *Tivoli Business Systems Manager Installation and Configuration Guide*.

The following table indicates the return codes from the EDI:

Code	Description
0	Normal Completion
4	EDI not available (Object Pump not up or not running EDI)
8	NOSPACE (Not enough space in the Object Pump EDI buffer for data)
12	Data item is larger than maximum allowed by EDI (1024 bytes)
16	GETMAIN for working storage failed (PC routine)
20	GETMAIN for working storage failed (EDI routine)
24	An abend occurred, code in register 1 (EDI routine)
28	An abend occurred, code in register 1 (PC routine)
32	Zero length data passed (this does not occur because the module does not pass zero length data items to EDI)

Scenario Event Flow



1. Started Task (APPC) monitored by SA OS/390 issues a warning message *DCR005W* message generated.
2. The Application Capture within the external product or application formats message specific information for passing and calls the REXX program *GTMEDISA*.
3. REXX program *GTMEDISA* receives the message information from the Application Capture. The program formats the information and executes the call to EDI (*GTMRXEDI*) passing the reformatted message information.
4. EDI receives the passed message information, adds the date, time, and sequence number attributes to the message, and passes the information to the Object Pump *Trap Manager*.
5. The trapped message is sent via a LU 6.2 connection to Tivoli Business Systems Manager on NT for posting to the Started Task (APPC) object within Tivoli Business Systems Manager.

Integration with Tivoli Business Systems Manager

Tivoli Business Systems Manager directly interfaces with SA OS/390 through the implementation of the EDI, which enables an external application to pass text-based messages and exceptions directly into the Object Pump on the same MVS system. When a status for a resource managed by SA OS/390 changes, Tivoli Business Systems Manager notifies you of this status change by modifying the Tivoli Business Systems Manager object representing the resource in question. For example, if SA OS/390 notices a change in status for data type DFHSM from UP to DOWN, Tivoli Business Systems Manager is notified and updates the Tivoli Business Systems Manager object representing data type DFHSM to DOWN (Red alert). Tivoli Business Systems Manager also is notified of any automation activities, on behalf of objects, by automation routines calling the Tivoli Business Systems Manager EDI. This interface enables you to manage and monitor system resources from a single focal point - the Tivoli Business Systems Manager workstation.

The interface is invoked when the following processes occur:

- Status for a SA OS/390 managed resource changes.
- Automation routine notifies Tivoli Business Systems Manager, via the EDI, of an automation routine **starting** for a failed object.
- Automation routine notifies Tivoli Business Systems Manager, via the EDI, of automation **ending** after a failed object has returned to a normal state.
- The passing of a Tivoli Business Systems Manager message or exception ID

Status Change

Within SA OS/390 application when a status change occurs for a resource the following message is generated:

A0F571I time : resname SUBSYSTEM STATUS FOR JOB job_name IS status - text

This message displays the new status of the resource.

- The variable **time** is the time the message was generated
- The variable **resname** is the name of the resource
- The variable **job_name** is the name of the job
- The variable **status** is the new status of the resource
- The variable **text** is additional information

When a status for a SA OS/390 managed resource changes, the SA OS/390 User Exit AOFEXSTA (Application Capture code) is invoked. The following code is part of the User Exit:

```

*           Parse the environment to obtain
/*      Object Type, Object Name, Msg/Except ID, SMF ID          */
Parse Arg restype ',' resource ',' status ',' sysname . /* @02C  */
/* -----Pass data to Tivoli Business Systems Manager ----- */
/* Following is how the data will be passed:                    */
/* ----- */
'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' TOEKN END_OF_DATA TRACEOFF
/* ----- */
Exit
/* ----- */
```

In the above example, the compiled REXX code, GTMEDISA resides in Tivoli Business Systems Manager's REXX dataset. The Tivoli Business Systems Manager REXX dataset has to be part of SA OS/390's **DSICLD DD** concatenation.

The following table contains the 26 different automation statuses that may be applied to applications under the control of SA OS/390:

ABENDING	CTLDOWN	HALTED	STOPPED
ACTIVE	DOWN	INACTIVE	STOPPING
ASSIST	ENDED	MOVED	STUCK
AUTODOWN	ENDING	RESTART	UP
AUTOTERM	EXTSTART	RUNNING	ZOMBIE
BREAKING	FALLBACK	STARTED	
BROKEN	HALFDOWN	STARTED2	

Once the EDI receives the necessary data, it is passed to TBSM Source/390 Object Pump via TRAPS. The SA OS/390 User Exit sends data with a prefixed message number, **GTMSA001**. The TBSM Source/390 Object Pump startup parameters module would contain the following code:

```
SA390_JOBNAME=INGESO
```

The previously mentioned code line activates a WTO trap within the TBSM Source/390 Object Pump as it comes up.

Once the TBSM Source/390 Object Pump has the data, it is sent to TBSM Source/390 Object Server (running on OS/390). The TBSM Source/390 Object Server then transmits (via LU 6.2 or the TCP/IP connection) to the NT Server.

Error Handling

When SA OS/390 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:

```
GTMSA101 EDI FAILURE ON sysname. RC=?? AC=???????? CN (console_name)
```

```
GTMSA102 EDI PROBLEM RESOLVED
```

sysname

system (OS) where EDI failure occurred

AB Abend Code from the EDI

CN (console_name)

name of the console where the message came from (for example, SA OS/390 NetView console)

RC Return Code from the EDI

Automation (Starting/Ending)

The automation process is very similar to the previously mentioned interface. The only difference is that instead of the User Exit AOFEXSTA calling the EDI program, the EDI is called from an automation routine. The automation routine then passes similar information as the REXX routine does in the status change process.

Note: The automation process (starting and ending) is **only** available for the Java-based console.

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

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

Following is an example of notifying Tivoli Business Systems Manager that automation is in progress of 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
```

Following is an example of notifying Tivoli Business Systems Manager 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
```

Following is an example of notifying Tivoli Business Systems Manager 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
```

Following are valid Format Type/Action Type/Data Type for the EDI:

Format Type

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

Action Type

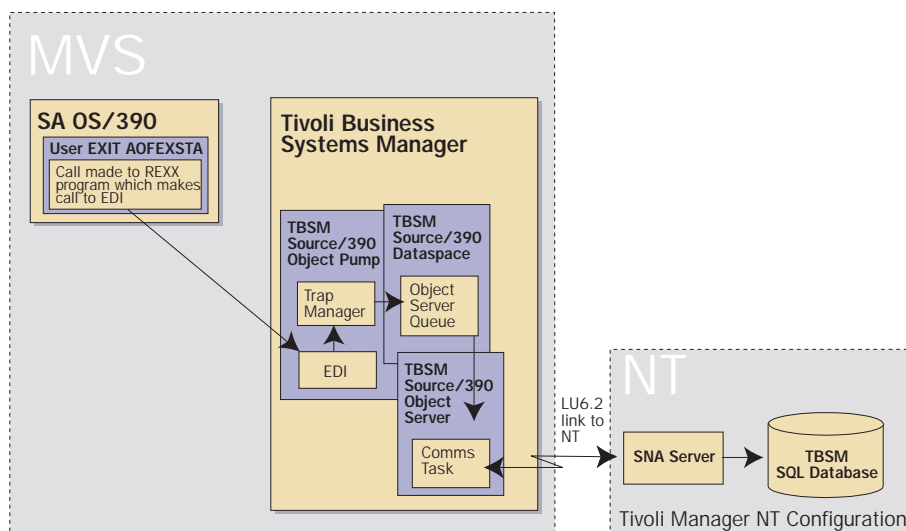
- 1 (STATECHANGE)
- 2 (MESSAGE)
- 3 (EXCEPTION)
- 4 (OPC MESSAGE)
- 5 (OPC EXCEPTION)

Data Type

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

Architecture

Tivoli Business Systems Manager is a suite of applications that resides in a three-tiered client/server environment and has server components on the MVS and Windows NT Server environments. When there is a status change for a SA OS/390 managed resource, the SA OS/390 USER EXIT (**AOFEXSTA**) passes the variables to a REXX routine. The REXX code builds the records with the necessary data (resource name, resource type, status, and system name) and pass it to the EDI, who in turn passes it to the *TBSM Source/390 Object Pump* (Object Pump). Once the Object Pump has the necessary data, it passes the data to the TBSM Source/390 Dataspace. The trapped data is passed to the TBSM Source/390 Object Server and finally to the NT Server.



Event Flow

1. The SA OS/390 USER EXIT **AOFEXSTA** passes the variables to a REXX routine
2. From the REXX program, a call is made to the EDI in order to pass the essential data (resource name, resource type, etc.)
3. The EDI formats the data as a WTO message and pass it to the Object Pump via WTO traps

-
4. The Object Pump passes the data to the Dataspace and it is placed on the Object Server Queue
 5. The Tivoli Business Systems Manager Server forwards the message to the SNA Server
 6. The SNA Server forwards the event to the Tivoli Business Systems Manager SQL database.

Note: When you are using *AOC/MVS OPC* automation within SA OS/390, you must use **EVJEXSTA** as your USER EXIT instead of **AOFEXSTA**.

Object Discovery

The discovery, extraction, and registration of all relevant system resources within an Enterprise is a critical step in the implementation of Tivoli Business Systems Manager. The identification process, also known as *Object Discovery*, uses SA OS/390 data as the source for the Discovery process.

Using the *Automation Control File* (ACF) as input, the data pertaining to all subsystems within the customer environment, including *Subsystem Name*, *Subsystem Description*, *Job Name*, *Application Groups* are discovered and extracted from the ACF.

These discovered and extracted objects are used for object registration within Tivoli Business Systems Manager.

Processing Elements

Data

Automation Control File

The Automation Control File (ACF), a product of SA OS/390, is a textual file that contains data pertaining to all subsystems within the Enterprise.

Using the Subsystem Name as the key, the ACF is examined to obtain the system components related to that Subsystem.

The combination of the data from the ACF file enables the population of the Enterprise objects within Tivoli Business Systems Manager.

Data Samples

Automation Control File

The ACF represents several types of information regarding the SA/OS390 environments. As such, there is no standard record layout for the elements within the file.

These samples represent the portions of the ACF that are relevant to the discovery and extraction processes.

File Header Information

```
1=----> MEMBER $BLDRPT FROM ACFFILE
/*****/
/*                                           */
/* BUILD Command Progress Display and Analysis Report */
/*                                           */
```

```

/* Function : BUILDF */
/* */
/* Enterprise: ABC_INSURANCE_COMPANY */
/* */
/* PolicyDB */
/* Name : PROD */
/* Data set: 'AOC.WXX.POLICY' */
/* Output : 'AOC.WXX.SOCNTL' */
/* */
/* User : DLU100 */
/* Date/Time : 99/08/31 at 18:04 */
/* */
/*****
/*****
/*
/* Command Progress Display Messages:
/*
/*****
Starting..... Automation Control File (ACF) generation
Starting..... Resynch for ADF (Application Defaults) entries
..... Resynch complete for ADF entries; no problems found.
Building..... ADF (Application Defaults) ACF fragments
..... Fragment Z999AADF built for ADF APPL_DEFAULTS_I802
..... Fragment Z997AADF built for ADF APPLICATION_DEFAULTS
Starting..... Resynch for AMC (Auto Msg Classes) entries
..... Resynch complete for AMC entries; no problems found.
Building..... AMC (Auto Msg Classes) ACF fragments
..... Fragment Z998AAMC built for AMC AUTO_MSG_CLASSES
Starting..... Resynch for AOP (Auto Operators) entries
..... Resynch complete for AOP entries; no problems found.
Building..... AOP (Auto Operators) ACF fragments
..... Fragment Z986AAOP built for AOP AUTO_OPERATORS
..... Fragment Z997AAOP built for AOP AUTO_OPERATORS_DB
..... Fragment Z99#AAOP built for AOP B301_AUTO_OPERATORS
..... Fragment Z98YAAOP built for AOP B801_AUTO_OPERATORS
*** End of Data

```

Operating System (Image) Data

```

==> MEMBER AOFACFMP FROM ACFFILE
* AOFACFMP member
*
* Generated at 18:04:26 on 99/08/31
*
* Policy Database: PROD
* Enterprise : ABC_INSURANCE_COMPANY
*
* SYSNAME      ACFNAME  RODMLoad  USE * SYSTEM NAME
*-----
E001      ACFZ99U  LOADZ99U  T * E001
E002      ACFZ99R  LOADZ99R  T * E002
E003      ACFZ998  LOADZ998  T * E003
E004      ACFZ99K  LOADZ99K  T * E004
E007      ACFZ99L  LOADZ99L  T * E007
E008      ACFZ990  LOADZ990  T * E008
E103      ACFZ98N  LOADZ98N  T * E103
E201      ACFZ99C  LOADZ99C  T * E201
E202      ACFZ98R  LOADZ98R  T * E202
E302      ACFZ996  LOADZ996  T * E302
E303      ACFZ992  LOADZ992  T * E303
E402      ACFZ995  LOADZ995  T * E402
E403      ACFZ98F  LOADZ98F  T * E403
E404      ACFZ98B  LOADZ98B  T * E404
E501      ACFZ991  LOADINS0  F * E501

```

Member Links

```
====-> MEMBER ACFZ998 FROM ACFFILE
* Title : Automation Control File
* Desc  : System Identifier : E003
* Built : DLU100 99/08/31 17:47:25 E003 NOICL
*-----
*Action  Member      Entry name      Status Short Description
%INCLUDE Z998ASYS    E003              BLT System E003
%INCLUDE Z950AAPL    ABENDAID          BLT ABENDAID
%INCLUDE Z95#AAPL    APPC              BLT APPC
%INCLUDE Z95$AAPL    ASCH              BLT ASCH
%INCLUDE Z949AAPL    BLSJPRMI          BLT BLSJPRMI
%INCLUDE Z947AAPL    CSAA              BLT CSAA
%INCLUDE Z94ZAAPL    FTP              BLT FTP
%INCLUDE Z94TAAPL    FTPSV01          BLT FTPSV01
%INCLUDE Z94RAAPL    FTPSV02          BLT FTPSV02
%INCLUDE Z92NAAPL    FTPSV03          BLT FTPSV03
%INCLUDE Z9WVAAPL    FTPSV04          BLT FTPSV04
%INCLUDE Z92TAAPL    DB03DBM1         BLT DB03DBM1
%INCLUDE Z92SAAPL    DB03DIST         BLT DB03DIST
```

Member Detail

```
====-> MEMBER Z92TAAPL FROM ACFFILE
*- Z92TAAPL -----
* Title : Application Details Policy
* Desc  : DB03DBM1
* Built : DLU100 99/08/31 16:54:09 DB03DBM1
*-----
*
SUBSYSTEM DB03DBM1,
    PARENT=(DB03MSTR),
    STARTCMD=PARENT,
    SHUTOPTIONS=PARENT,
    JOB=DB03DBM1,
    SDESC='DB03DBM1'
*
* NO THRESHOLDS DEFINED FOR : DB03DBM1
*
* NO AUTOMATION FLAG DETAILS SPECIFIED FOR APPLICATION : DB03DBM1
*
* NO AUTO MESSAGES DEFINED FOR : DB03DBM1
*
* SYSTEM ASSOCIATION INFORMATION FOR APPLICATION : DB03DBM1
* ASSOCIATION DB03DBM1,
    SECONDARY=(D903S,E302S)
* End of Run at 99/08/31 16:54:09
```

Discovery and Extraction Methodology

The methodology for the discovery and extraction of the objects within the ACF requires several distinct operations or steps.

The following documentation provides both the sequence of the steps to be performed and a description of those steps. In addition, several operations create intermediate data files. These are also detailed within each specific step.

Step 1: SYSNAME Pointer Capture

This step is the starting point for automated discovery. The data elements captured within this step represent the keys to that discovery process, and are posted to an intermediate file for use later in the discovery process.

Starting at the top of the **ACF**, examine each line of the file until this header record is encountered:

```
* SYSNAME      ACFNAME  RODMLOAD  USE * SYSTEM NAME
```

Of interest to the discovery and extraction process are the values for SYSNAME and ACFNAME. SYSNAME refers to the Operating System (or Image) name, while ACFNAME provides linkage to the Member Name used in Step 2.

ACF Data example:

```
* SYSNAME      ACFNAME  RODMLOAD  USE * SYSTEM NAME
*-----*
  B301          ACFZ99A  LOADZ99A  T  * B301
  B801          ACFZ98M  LOADZ98M  T  * B801
  E003          ACFZ998  LOADZ998  T  * E003
```

Examine the records following the header record for any records that do not begin with a comment character (*). When discovered, extract the SYSNAME & ACFNAME values from the ACF, placing those data element into an intermediate file called the Image Header file.

Image Header File example:

```
SYSNAME      ACFNAME
-----
B301          ACFZ99A
B801          ACFZ98M
E003          ACFZ998
```

The discovery continues to the next line with each discovered SYSNAME and ACFNAME value being written to the Image Header file.

This processing step terminates when '*****End of Data**' is encountered.

Note: Only one SYSNAME & ACFNAME header should be found.

Proceed to Member Links Capture (Step 2).

Note: ***Bold Italic*** is used to identify the data elements that are used for subsequent processing detailed with the document.

Step 2: Member Links Capture

Building on the data collected in Step 1, this processing provides information as to the members associated with a specific ACFNAME. This discovered data is written to an intermediate file for use further in the discovery process.

Starting at the top of the ACF, use the ACFNAME value that was extracted and written to the Image Header file in Step 1 as a search argument to obtain the Member Links associated with that ACFNAME.

The search pattern format for this search is:

```
====-> MEMBER <ACFNAME value> FROM ACFFILE
```

ACF Data example:

```
====-> MEMBER ACFZ998 FROM ACFFILE
* Title : Automation Control File
* Desc  : System Identifier : E003
* Built : DLU100 99/08/31 17:47:25 E003 NOICL
```

```

*-----
*
*Action  Member      Entry name      Status Short Description
*-----
%INCLUDE Z998ASYS    E003              BLT System E003
%INCLUDE Z9$YAAPL    CLASS_HSM_01  BLT Class 01 fjr HSM
%INCLUDE Z92TAAPL    DB03DBM1      BLT DB03DBM1

```

Examine the records for any records that do not begin with a comment character (*). When discovered, extract the Member value from the ACF.

The extracted Member value, in conjunction with the SYSNAME and ACFNAME values from the Image Header file (created in Step 1) are written to another intermediate file called the Anchor Member file.

Anchor Member File example:

ACFNAME	SYSNAME	MEMBER
ACFZ998	E003	Z998ASYS
ACFZ998	E003	Z9\$YAAPL
ACFZ998	E003	Z92TAAPL

For each of the following '%INCLUDE' lines record the discovered Member, SYSNAME and ACFNAME values in the Anchor Member file.

The collection process for this particular ACFNAME value terminates when the **'*End of Run'** is encountered.

Then this step is repeated, starting at the top of the ACF using the next Member value.

Proceed to Extract Member Detail Capture (Step 3).

Step 3: Member Detail Capture

Using the data captured in Step 2 for selection, this step provides further information including Job Name, Job Description and secondary Operating System (Image) associated with those discovered Members. Once captured, this information is also posted to an intermediate file for future use.

Starting from the top of the ACF and utilizing the intermediate Anchor Member file created in Step 2, use the Member value as a search argument to be used against the ACF.

The search pattern format for this search is:

```

==--> MEMBER <Member value> FROM ACFFILEACF Data example:
==--> MEMBER Z92TAAPL FROM ACFFILE
*- Z92TAAPL -----
* Title : Application Details Policy
* Desc  : DB03DBM1
* Built : DLU100 99/08/31 16:54:09 DB03DBM1
*-----
*
SUBSYSTEM DB03DBM1,
  PARENT=(DB03MSTR),
  STARTCMD=PARENT,
  SHUTOPTIONS=PARENT,
  JOB=Z92TAAPL,
  SDESC='Z92TAAPL'
*
* NO THRESHOLDS DEFINED FOR : DB03DBM1

```

```

*
* NO AUTOMATION FLAG DETAILS SPECIFIED FOR APPLICATION : DB03DBM1
*
* NO AUTO MESSAGES DEFINED FOR : DB03DBM1
*
* SYSTEM ASSOCIATION INFORMATION FOR APPLICATION : DB03DBM1
*
ASSOCIATION DB03DBM1,
SECONDARY=(D903S,E302S)

```

The values of Member, JOB, SDESC (short description), DESC (long description - if present) and SECONDARY are extracted from the ACF.

The last character of each Image from the SECONDARY value list as delimited by commas is removed from that value, leaving just the Image Name. In addition, if the current Member value's Image is found in the Secondary value list, the record needs to be modified to reflect its secondary nature. This is done by taking JOB, SDESC and DESC and appending "SECONDARY" to their data. The only exception to this is if the individual field is equal to null. That field should stay equal to null. This does not affect the modification of the other fields.

Member, SDEC, DESC, and JOB and SYSNAME from the intermediate file are then written to a final intermediate file known as the Member Detail file.

Member Detail File example:

Member	Description Short	Description Long	Job	SYSNAME
----- Z92TAAPL DB03DBM1				
Z92TAAPL	DB03DBM1	SECONDARYDB03DBM1	SECONDARY	D903
Z92TAAPL	DB03DBM1	SECONDARYDB03DBM1	SECONDARY	E302

This step is repeated, starting at the top of the ACF using the next Member value. As this step executes, other discovered JOB, SDESC, DESC and SECONDARY values are written to the Member Detail file with the corresponding Member Name and SYSNAME values.

Each member instance of Member Detail capture is terminated when '***End of Run**' is encountered.

Proceed to Tivoli Business Systems Manager Loader File Creation (Step 4).

Step 4: Tivoli Business Systems Manager Loader File Creation

This next step formats the discovered information stored in the intermediate files. This file is applied against the customer's job naming standards in order to assign Object Types to this formatted data. Object Types include CICS, IMS, STC, IDMS, and others.

Starting from the top of the Member Detail file, extract the values of SYSNAME, JOB, SDESC and DESC for each record. In the event that the discovered DESC value is NULL, the DESC value is set to the value of SDESC. These extracted values are written to a file known as the Pre-Load File.

The Pre-Load File is textual in nature and backslash-delimited.

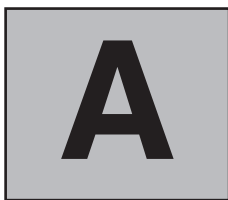
Pre-Load File Record Layout

```
<SYSNAME>\Object Type\<JOB>\<DESC>
```

Pre-Load File example:

```
E003\\DB03DBM1\DB03DBM1  
D903\\DB03DBM1 SECONDARY\DB03DBM1 SECONDARY  
E302\\DB03DBM1 SECONDARY\Execute after 10PM SECONDARY
```

This step terminates with the end-of-file marker for the Member Detail file is encountered.



Status Messages for TBSM Objects

Incoming SA OS/390 State Message	TBSM Object Desired State	Initial TBSM Object Anticipated Current State (prior to message)	Resulting TBSM Object Current State	TBSM Message Alert State	TBSM Message Priority	Comments
UP Statuses		The following statuses indicate that the application monitoring status should be Active.				
ACTIVE	Inactive	Inactive	Starting (if Starting not supported -Active)	Yellow	High	Application is running, but not yet ready for work. It is anticipated that this message will not be received while in this desired state. If so a schedule violation should be generated.
	Active	Inactive	Starting (if Starting not supported -Active)	Yellow	High	
RUNNING	Inactive	Inactive	Active	Yellow	High	This status is equivalent to UP but is used for transient applications.
	Active	Inactive	Active	Yellow	High	

Incoming SA OS/390 State Message	TBSM Object Desired State	Initial TBSM Object Anticipated Current State (prior to message)	Resulting TBSM Object Current State	TBSM Message Alert State	TBSM Message Priority	Comments
HALTED*	Inactive	Active	Active	Yellow	High	Application is still running, however something has occurred that has impacted its capabilities. It is anticipated that this message will not be received while in this desired state. If so a schedule violation should be generated.
	Active	Active	Active	Yellow	High	
UP*	Inactive	Starting Inactive	Active	Green	High	This is the final message signaling the application is up and running. * It is anticipated that this message will not be received while in this desired state. If so a schedule violation should be generated.
	Active	Starting Inactive	Active	Green	Ignore	N/A
DOWN Statuses		The following statuses indicate that the application monitoring status should be Inactive.				
AUTODOWN*	Inactive	Active	Stopping (If Stopping not supported stay Active)	Green	High	The application is shut down.
	Active	Active	Stopping (If Stopping not supported stay Active)	Green	Ignore	This should generate an event other than Green High if something is being Shutdown outside of scheduled hours. If so a schedule violation should be generated.

Incoming SA OS/390 State Message	TBSM Object Desired State	Initial TBSM Object Anticipated Current State (prior to message)	Resulting TBSM Object Current State	TBSM Message Alert State	TBSM Message Priority	Comments
BROKEN*	Inactive	Active	Abended	N/A	N/A	Suffered a non-recoverable error and SA will not restart it. SA may be instructed to a set of predefined attempts at restarting, however after these have been exhausted, this message is sent.
	Active	Active	Abended	Red	High	
CTLDOWN*	Inactive	Inactive	Inactive	Green	Ignore	The application is shut down and SA is not allowed to restart it.
	Active	Inactive	Active	Green	Ignore	???? Schedule Violation???
DOWN*	Inactive	Inactive	Inactive	Ignore	Green	The application has not been started during the lifetime of this SA/390.
	Active	Inactive	Inactive	Green	Ignore	At SA startup, the application has state of the application has been determined as down. Must generate a schedule violation exception, which should change alert state to Red.

Incoming SA OS/390 State Message	TBSM Object Desired State	Initial TBSM Object Anticipated Current State (prior to message)	Resulting TBSM Object Current State	TBSM Message Alert State	TBSM Message Priority	Comments
ENDED*	Inactive	Active	Inactive	Green	Ignore	This status is for transient applications only. For example, if an address space is intended to start perform a task and then end.
	Active	Active	Inactive	Green	Ignore	See above.
FALLBACK	Inactive	Inactive	Inactive	Green	Ignore	The application is not running: this is one of its secondary systems: it should be active on another system. If the other system fails, the application can fall back to (restart on) this system.
	Active	Inactive	Inactive	Green	Ignore	
INACTIVE *	Inactive	Active Starting	Inactive	Yellow	High	At some point in monitoring, the application monitor status was INACTIVE when the automation status indicated that it should be either ACTIVE or STARTING.
	Active	Active Starting	Inactive	Yellow	High	
MOVED	Inactive	Inactive	Inactive	Green	Ignore	The application is not running: this is one of its primary systems: it should be active on another system.
	Active	Inactive	Inactive	Green	Ignore	

Incoming SA OS/390 State Message	TBSM Object Desired State	Initial TBSM Object Anticipated Current State (prior to message)	Resulting TBSM Object Current State	TBSM Message Alert State	TBSM Message Priority	Comments
RESTART	Inactive	Abended Inactive	Starting (If Starting not supported stay Inactive)	Green	Ignore	The application is ready to be started. It has been previously active in the system.
	Active	Abended Inactive	Starting (If Starting not supported stay Inactive)	Green	Ignore	
STOPPED*	Inactive	Active	Inactive	Green	High	The application has been shut down by an external agent such as an operator console. SA cannot restart it and will not allow the MVS Automatic Restart Manager to restart it. It is anticipated that this message will not be received while in this desired state. If so a schedule violation should be generated.
	Active	Active	Inactive	Red	High	
STARTING Statuses		The following statuses indicate that the application monitoring status may be Starting or Active.				
EXTSTART	Inactive	Inactive Abended	Starting (Inactive where starting not supported)	Green	Ignore	SA OS/390 has determined that the application is being started or restarted by an agent external to SA OS/390.
	Active	Inactive Abended	Starting (Inactive Where starting not supported)	Green	Ignore	

Incoming SA OS/390 State Message	TBSM Object Desired State	Initial TBSM Object Anticipated Current State (prior to message)	Resulting TBSM Object Current State	TBSM Message Alert State	TBSM Message Priority	Comments
STARTED	Inactive	Inactive Abended	Starting (Inactive where starting not supported)	Green	Ignore	The commands to start the application have been issued, but it has yet to start running.
	Active	Inactive Abended	Starting (Inactive where starting not supported)	Green	Ignore	
STARTED2*	Active	Inactive Abended	Starting (Inactive where starting not supported)	Yellow	High	The application has become active, but has not indicated that it is ready to accept work within its start delay.
	Inactive	Inactive Abended	Starting (Inactive where starting not supported)	Yellow	High	
TRANSITIONAL Statuses		The following statuses are either a part of the shutdown process, or indicate that the application may or may not be present in the machine. ALL Application monitoring statuses are possible with any transitional status.				
ABENDING	Inactive	Active	Abended			The application is undergoing a recoverable abnormal end. The status is entered when a termination messages with ABEND=YES attribute is received. This is a case that should not be handled identical based on differing Desired States.
	Active	Active	Abended	Red	High	

Incoming SA OS/390 State Message	TBSM Object Desired State	Initial TBSM Object Anticipated Current State (prior to message)	Resulting TBSM Object Current State	TBSM Message Alert State	TBSM Message Priority	Comments
ASSIST	Inactive	Active	Active	Green	Ignore	Unusual status. When an application has ASSIST=DISPLAY coded for one or more of its automation flags, SA OS/390 puts the application into ASSIST status whenever it wants to issue a command that is controlled by that automation flag.
	Active	Active	Active	Green	Ignore	
AUTOTERM	Inactive	Active	Stopping (If Stopping not supported remain Active)	Green	Ignore	SA OS/390 is in the process of shutting the application down.
	Active	Active	Stopping (If Stopping not supported remain Active)	Green	Ignore	
BREAKING	Inactive	Active	Active	Yellow	High	This application is undergoing a non-recoverable abend.
	Active	Active	Active	Yellow	High	

Incoming SA OS/390 State Message	TBSM Object Desired State	Initial TBSM Object Anticipated Current State (prior to message)	Resulting TBSM Object Current State	TBSM Message Alert State	TBSM Message Priority	Comments
ENDING	Inactive	Active	Stopping (If Stopping not supported remain Active)	Green	Ignore	A transient application is in the process of terminating. A transient application goes to ENDING status when a termination message is received for it, and it is not being shut down by SA OS/390.
	Active	Active	Stopping (If Stopping not supported remain Active)	Green	Ignore	
HALFDOWN	Inactive	Active	Active	Yellow	High	SA OS/390 was in the process of shutting down an application down, but the shutdown was canceled while it was in progress.
	Active	Active	Active	Yellow	High	
STOPPING	Inactive	Active	Stopping (If Stopping not supported remain Active)	Green	Ignore	The application is being shut down by an external agent. This status is entered when a TERMINATION message is received while SA OS/390 is not in the process of shutting the application down.
	Active	Active	Stopping (If Stopping not supported remain Active)	Green	Ignore	

Incoming SA OS/390 State Message	TBSM Object Desired State	Initial TBSM Object Anticipated Current State (prior to message)	Resulting TBSM Object Current State	TBSM Message Alert State	TBSM Message Priority	Comments
STUCK*	Active	Active	Active	Yellow	High	This message is sent if all SA 390 shutdown commands have been exhausted and the application remains UP.
	Inactive	Active	Active	Yellow	High	NA - Schedule violation if occurs.
ZOMBIE*	Inactive	Active	Unknown	Yellow	High	NA
	Active	Active	Unknown	Yellow	High	If an application has indicated that its final termination message has been received, yet SA still finds the application. The application is put into a ZOMBIE status if the situation persists for more than twice the application termination time.

B

Messages for Registered Object Types

Batch

Class ID	AOF State	TBSM State	Alert State	Priority
BATC	ABENDING		YELLOW	HIGH
BATC	ACTIVE	STARTING	GREEN	IGNORE
BATC	ASSIST		GREEN	IGNORE
BATC	AUTODOWN	INACTIVE	GREEN	IGNORE
BATC	AUTOTERM	STOPPING	GREEN	IGNORE
BATC	BREAKING		YELLOW	HIGH
BATC	BROKEN	ABENDED	RED	HIGH
BATC	CTLDOWN	INACTIVE	GREEN	IGNORE
BATC	DOWN	INACTIVE	GREEN	IGNORE
BATC	ENDED		GREEN	IGNORE
BATC	ENDING		GREEN	IGNORE
BATC	EXTSTART	STARTING	GREEN	IGNORE
BATC	FALLBACK		GREEN	IGNORE
BATC	HALFDOWN		YELLOW	HIGH
BATC	HALTED		YELLOW	HIGH
BATC	INACTIVE	INACTIVE	YELLOW	HIGH
BATC	MOVED		GREEN	IGNORE
BATC	RESTART	STARTING	GREEN	IGNORE
BATC	RUNNING	RUNNING	GREEN	HIGH
BATC	STARTED	STARTING	GREEN	IGNORE
BATC	STARTED2	STARTING	YELLOW	HIGH
BATC	STOPPED	INACTIVE	GREEN	IGNORE
BATC	STOPPING	STOPPING	GREEN	IGNORE
BATC	STUCK		YELLOW	HIGH
BATC	UP	RUNNING	GREEN	HIGH
BATC	ZOMBIE		YELLOW	HIGH

CICS

Class ID	AOF State	TBSM State	Alert State	Priority
CICS	ABENDING		YELLOW	HIGH
CICS	STARTING	ACTIVE	GREEN	IGNORE
CICS	ASSIST		GREEN	IGNORE
CICS	AUTODOWN	INACTIVE	GREEN	IGNORE
CICS	AUTOTERM	STOPPING	GREEN	IGNORE
CICS	BREAKING		YELLOW	HIGH
CICS	BROKEN	ABENDED	RED	HIGH
CICS	CTLDOWN	INACTIVE	GREEN	IGNORE
CICS	DOWN	INACTIVE	GREEN	IGNORE
CICS	ENDED		GREEN	IGNORE
CICS	ENDING		GREEN	IGNORE
CICS	EXTSTART	STARTING	GREEN	IGNORE
CICS	FALLBACK		GREEN	IGNORE
CICS	HALFDOWN		YELLOW	HIGH
CICS	HALTED		YELLOW	HIGH
CICS	INACTIVE	INACTIVE	YELLOW	HIGH
CICS	MOVED		GREEN	IGNORE
CICS	RESTART	STARTING	GREEN	IGNORE
CICS	RUNNING	ACTIVE	GREEN	HIGH
CICS	STARTED	STARTING	GREEN	IGNORE
CICS	STARTED2	STARTING	YELLOW	HIGH
CICS	STOPPED	INACTIVE	GREEN	IGNORE
CICS	STOPPING	STOPPING	GREEN	IGNORE
CICS	STUCK		YELLOW	HIGH
CICS	UP	ACTIVE	GREEN	HIGH
CICS	ZOMBIE		YELLOW	HIGH

CMAS

Class ID	AOF State	TBSM State	Alert State	Priority
CMAS	ABENDING		YELLOW	HIGH
CMAS	ACTIVE	STARTING	GREEN	IGNORE
CMAS	ASSIST		GREEN	IGNORE
CMAS	AUTODOWN	INACTIVE	GREEN	IGNORE
CMAS	AUTOTERM		GREEN	IGNORE
CMAS	BREAKING		YELLOW	HIGH
CMAS	BROKEN	ABENDED	RED	HIGH
CMAS	CTLDOWN	INACTIVE	GREEN	IGNORE
CMAS	DOWN	INACTIVE	GREEN	IGNORE

Class ID	AOF State	TBSM State	Alert State	Priority
CMAS	ENDED		GREEN	IGNORE
CMAS	ENDING		GREEN	IGNORE
CMAS	EXTSTART	STARTING	GREEN	IGNORE
CMAS	FALLBACK		GREEN	IGNORE
CMAS	HALFDOWN		YELLOW	HIGH
CMAS	HALTED		YELLOW	HIGH
CMAS	INACTIVE	INACTIVE	YELLOW	HIGH
CMAS	MOVED		GREEN	IGNORE
CMAS	RESTART	STARTING	GREEN	IGNORE
CMAS	RUNNING	ACTIVE	GREEN	HIGH
CMAS	STARTED	STARTING	GREEN	IGNORE
CMAS	STARTED2	STARTING	YELLOW	HIGH
CMAS	STOPPED	INACTIVE	GREEN	IGNORE
CMAS	STOPPING		GREEN	IGNORE
CMAS	STUCK		YELLOW	HIGH
CMAS	UP	ACTIVE	GREEN	HIGH
CMAS	ZOMBIE		YELLOW	HIGH

DB2 PM

Class ID	AOF State	TBSM State	Alert State	Priority
DB2PM	ABENDING		YELLOW	HIGH
DB2PM	ACTIVE	STARTING	GREEN	IGNORE
DB2PM	ASSIST		GREEN	IGNORE
DB2PM	AUTODOWN	INACTIVE	GREEN	IGNORE
DB2PM	AUTOTERM		GREEN	IGNORE
DB2PM	BREAKING		YELLOW	HIGH
DB2PM	BROKEN	ABENDED	RED	HIGH
DB2PM	CTLDOWN	INACTIVE	GREEN	IGNORE
DB2PM	DOWN	INACTIVE	GREEN	IGNORE
DB2PM	ENDED		GREEN	IGNORE
DB2PM	ENDING		GREEN	IGNORE
DB2PM	EXTSTART	STARTING	GREEN	IGNORE
DB2PM	FALLBACK		GREEN	IGNORE
DB2PM	HALFDOWN		YELLOW	HIGH
DB2PM	HALTED		YELLOW	HIGH
DB2PM	INACTIVE	INACTIVE	YELLOW	HIGH
DB2PM	MOVED		GREEN	IGNORE
DB2PM	RESTART	STARTING	GREEN	IGNORE
DB2PM	RUNNING	ACTIVE	GREEN	HIGH
DB2PM	STARTED	STARTING	GREEN	IGNORE

Class ID	AOF State	TBSM State	Alert State	Priority
DB2PM	STARTED2	STARTING	YELLOW	HIGH
DB2PM	STOPPED	INACTIVE	GREEN	IGNORE
DB2PM	STOPPING		GREEN	IGNORE
DB2PM	STUCK		YELLOW	HIGH
DB2PM	UP	ACTIVE	GREEN	HIGH
DB2PM	ZOMBIE		YELLOW	HIGH

DB2 Subsystem

Class ID	AOF State	TBSM State	Alert State	Priority
DB2S	ABENDING		YELLOW	HIGH
DB2S	ACTIVE		GREEN	IGNORE
DB2S	ASSIST		GREEN	IGNORE
DB2S	AUTODOWN	STOPPED	GREEN	IGNORE
DB2S	AUTOTERM		GREEN	IGNORE
DB2S	BREAKING		YELLOW	HIGH
DB2S	BROKEN		RED	HIGH
DB2S	CTLDOWN	STOPPED	GREEN	IGNORE
DB2S	DOWN	STOPPED	GREEN	IGNORE
DB2S	ENDED		GREEN	IGNORE
DB2S	ENDING		GREEN	IGNORE
DB2S	EXTSTART		GREEN	IGNORE
DB2S	FALLBACK		GREEN	IGNORE
DB2S	HALFDOWN		YELLOW	HIGH
DB2S	HALTED		YELLOW	HIGH
DB2S	INACTIVE	STOPPED	YELLOW	HIGH
DB2S	MOVED		GREEN	IGNORE
DB2S	RESTART		GREEN	IGNORE
DB2S	RUNNING	RUNNING	GREEN	HIGH
DB2S	STARTED		GREEN	IGNORE
DB2S	STARTED2		YELLOW	HIGH
DB2S	STOPPED	STOPPED	GREEN	IGNORE
DB2S	STOPPING		GREEN	IGNORE
DB2S	STUCK		YELLOW	HIGH
DB2S	UP	RUNNING	GREEN	HIGH
DB2S	ZOMBIE		YELLOW	HIGH

IDMS

Class ID	AOF State	TBSM State	Alert State	Priority
IDMS	ABENDING		YELLOW	HIGH

Class ID	AOF State	TBSM State	Alert State	Priority
IDMS	ACTIVE	STARTING	GREEN	IGNORE
IDMS	ASSIST		GREEN	IGNORE
IDMS	AUTODOWN	INACTIVE	GREEN	IGNORE
IDMS	AUTOTERM	STOPPING	GREEN	IGNORE
IDMS	BREAKING		YELLOW	HIGH
IDMS	BROKEN	ABENDED	RED	HIGH
IDMS	CTLDOWN	INACTIVE	GREEN	IGNORE
IDMS	DOWN	INACTIVE	GREEN	IGNORE
IDMS	ENDED		GREEN	IGNORE
IDMS	ENDING		GREEN	
IDMS	EXTSTART	STARTING	GREEN	IGNORE
IDMS	FALLBACK		GREEN	IGNORE
IDMS	HALFDOWN		YELLOW	HIGH
IDMS	HALTED		YELLOW	HIGH
IDMS	INACTIVE	INACTIVE	YELLOW	
IDMS	MOVED		GREEN	IGNORE
IDMS	RESTART	STARTING	GREEN	IGNORE
IDMS	RUNNING	ACTIVE	GREEN	HIGH
IDMS	STARTED	STARTING	GREEN	IGNORE
IDMS	STARTED2	STARTING	YELLOW	HIGH
IDMS	STOPPED	INACTIVE	GREEN	IGNORE
IDMS	STOPPING	STOPPING	GREEN	IGNORE
IDMS	STUCK		YELLOW	HIGH
IDMS	UP	ACTIVE	GREEN	HIGH
IDMS	ZOMBIE		YELLOW	HIGH

IMS

Class ID	AOF State	TBSM State	Alert State	Priority
IMS	ABENDING		YELLOW	HIGH
IMS	ACTIVE		GREEN	IGNORE
IMS	ASSIST		GREEN	IGNORE
IMS	AUTODOWN	INACTIVE	GREEN	IGNORE
IMS	AUTOTERM		GREEN	IGNORE
IMS	BREAKING		YELLOW	HIGH
IMS	BROKEN	ABENDED	RED	HIGH
IMS	CTLDOWN	INACTIVE	GREEN	IGNORE
IMS	DOWN	INACTIVE	GREEN	IGNORE
IMS	ENDED		GREEN	IGNORE
IMS	ENDING		GREEN	IGNORE
IMS	EXTSTART		GREEN	IGNORE

Class ID	AOF State	TBSM State	Alert State	Priority
IMS	FALLBACK		GREEN	IGNORE
IMS	HALFDOWN		YELLOW	HIGH
IMS	HALTED		YELLOW	HIGH
IMS	INACTIVE	INACTIVE	YELLOW	HIGH
IMS	MOVED		GREEN	IGNORE
IMS	RESTART		GREEN	IGNORE
IMS	RUNNING	ACTIVE	GREEN	HIGH
IMS	STARTED		GREEN	IGNORE
IMS	STARTED2		YELLOW	HIGH
IMS	STOPPED	INACTIVE	GREEN	IGNORE
IMS	STOPPING		GREEN	IGNORE
IMS	STUCK		YELLOW	HIGH
IMS	UP	ACTIVE	GREEN	HIGH
IMS	ZOMBIE		YELLOW	HIGH

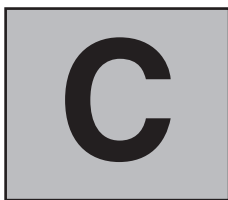
Omegamon

Class ID	AOF State	TBSM State	Alert State	Priority
OMEG	ABENDING		YELLOW	HIGH
OMEG	ACTIVE		GREEN	IGNORE
OMEG	ASSIST		GREEN	IGNORE
OMEG	AUTODOWN	INACTIVE	GREEN	IGNORE
OMEG	AUTOTERM		GREEN	IGNORE
OMEG	BREAKING		YELLOW	HIGH
OMEG	BROKEN	ABENDED	RED	HIGH
OMEG	CTLDOWN	INACTIVE	GREEN	IGNORE
OMEG	DOWN	INACTIVE	GREEN	IGNORE
OMEG	ENDED		GREEN	IGNORE
OMEG	ENDING		GREEN	IGNORE
OMEG	EXTSTART		GREEN	IGNORE
OMEG	FALLBACK		GREEN	IGNORE
OMEG	HALFDOWN		YELLOW	HIGH
OMEG	HALTED		YELLOW	HIGH
OMEG	INACTIVE	INACTIVE	YELLOW	HIGH
OMEG	MOVED		GREEN	IGNORE
OMEG	RESTART		GREEN	IGNORE
OMEG	RUNNING	ACTIVE	GREEN	HIGH
OMEG	STARTED		GREEN	IGNORE
OMEG	STARTED2		YELLOW	HIGH
OMEG	STOPPED	INACTIVE	GREEN	IGNORE
OMEG	STOPPING		GREEN	IGNORE

Class ID	AOF State	TBSM State	Alert State	Priority
OMEG	STUCK		YELLOW	HIGH
OMEG	UP	ACTIVE	GREEN	HIGH
OMEG	ZOMBIE		YELLOW	HIGH

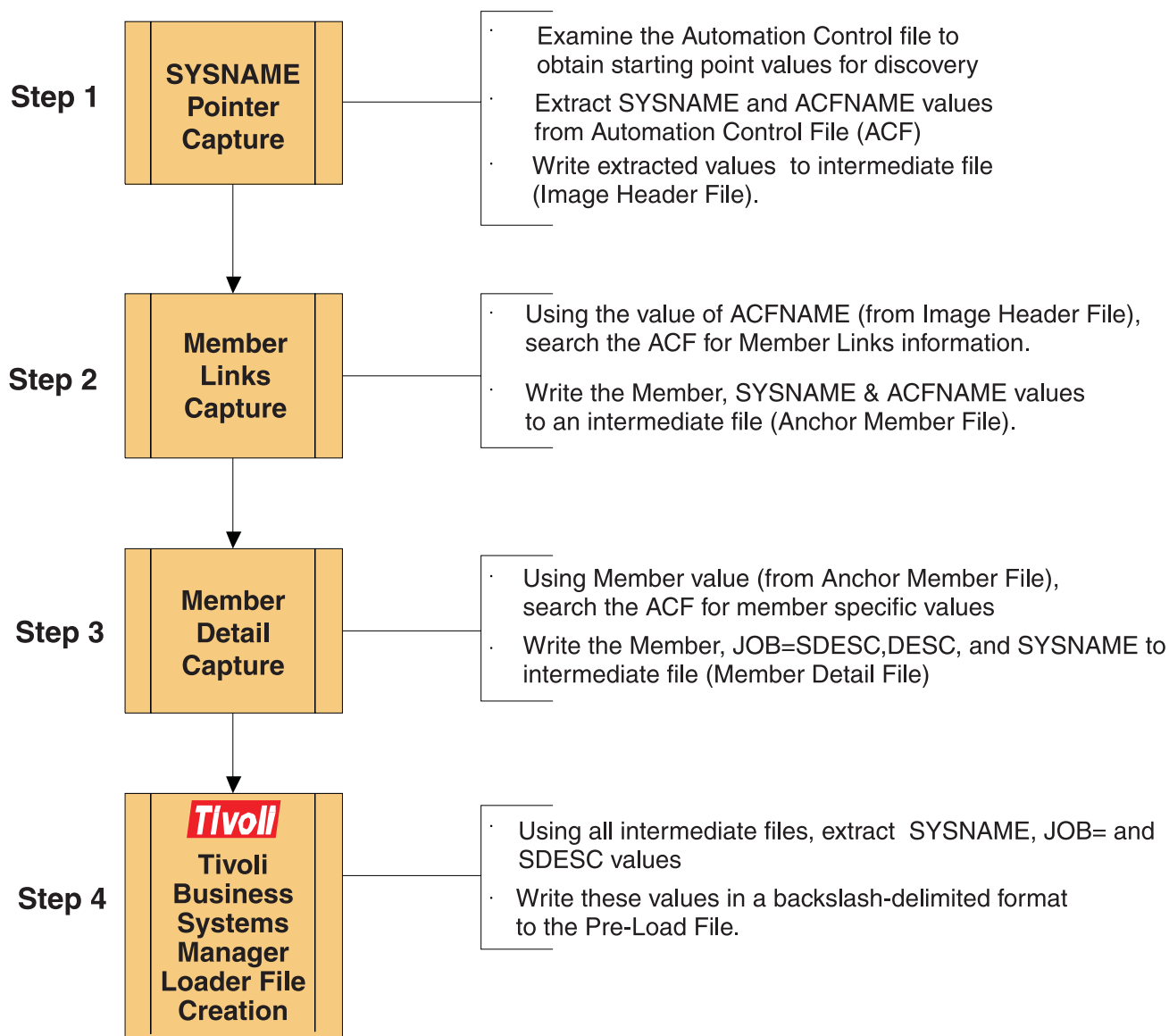
Started Task

Class ID	AOF State	TBSM State	Alert State	Priority
STC	ABENDING		YELLOW	HIGH
STC	ACTIVE	STARTING	GREEN	IGNORE
STC	ASSIST		GREEN	IGNORE
STC	AUTODOWN	INACTIVE	GREEN	IGNORE
STC	AUTOTERM	STOPPING	GREEN	IGNORE
STC	BREAKING		YELLOW	HIGH
STC	BROKEN	ABENDED	RED	HIGH
STC	CTLDOWN	INACTIVE	GREEN	IGNORE
STC	DOWN	INACTIVE	GREEN	IGNORE
STC	ENDED		GREEN	IGNORE
STC	ENDING		GREEN	IGNORE
STC	EXTSTART	STARTING	GREEN	IGNORE
STC	FALLBACK		GREEN	IGNORE
STC	HALFDOWN		YELLOW	HIGH
STC	HALTED		YELLOW	HIGH
STC	INACTIVE	INACTIVE	YELLOW	HIGH
STC	MOVED		GREEN	IGNORE
STC	RESTART	STARTING	GREEN	IGNORE
STC	RUNNING	ACTIVE	GREEN	HIGH
STC	STARTED	STARTING	GREEN	IGNORE
STC	STARTED2	STARTING	YELLOW	HIGH
STC	STOPPED	INACTIVE	GREEN	IGNORE
STC	STOPPING	STOPPING	GREEN	IGNORE
STC	STUCK		YELLOW	HIGH
STC	UP	ACTIVE	GREEN	HIGH
STC	ZOMBIE		YELLOW	HIGH



Object Discovery Process Diagrams

Automated SA/OS390 Bulk Object Discovery Process Flow



Data Flow Diagram

Bulk Discovery Data Flow Using the SA OS/390 Automation Control File





List of Files Used in Object Discovery

Automation Control File (ACF)

This file provides the data source for the Discovery process. This file referenced several different times within the processes.

Image Header File

Provides an intermediate file to store the results of Step 1 in Appendix A. This file is used within several other processes within discovery.

Anchor Member File

An intermediate file, this collection of Member-specific information is created in Step 2 in Appendix A. This data is also used within other portions of the discovery process.

Member Detail File

The final intermediate file contains the detail data as to the discovered Member. This data is used in additional discovery and registration processing.



Configuration of Event-Based Discovery

Discovery of resources managed by SA/390 can be automated in Tivoli Business Systems Manager by the configuration of a database table, *SA390ResourceTypeMap*. This section assumes that the reader is familiar with basic SQL concepts, and can issue database commands using a SQL command console (such as ISQL for Microsoft SQL Server).

SA390ResourceTypeMap Table

This table has the following columns:

Column Name	Data Type	Description
SMFID_PAT	VARCHAR(255)	SQL wildcard expression for matching the SMFID field of events. The combination of SMFID_PAT, RESOURCE_TYPE_PAT, and OBJECT_NAME_PAT must be unique.
RESOURCE_TYPE_PAT	VARCHAR(255)	SQL wildcard expression for matching the RESOURCE_TYPE field of events.
OBJECT_NAME_PAT	VARCHAR(255)	SQL wildcard expression for matching the OBJECT_NAME field of events.
cid	ClassID	Class id to be assigned to matching object. If null then matching records are discarded.
priority	INT	Relative priority of this row. Events are matched against rows with higher priority before those with lower priority.
enabled	TINYINT	If 1 (the default) then the row is used for pattern matching. If 0, then the row is not used.
parent_cid	ClassID	Optional value specifying class of object that is to be the parent of the discovered resource. This is typically 'OS' for Operating System. If this value is not used then the resource is placed under the OS that reported the event or possibly on another OS in the Sysplex if the resource already exists there and is of a type managed by Sysplex.
parent_id	ObjID	Optional value specifying the instance id of the object that is to be the parent of the discovered resource.

This table is initially populated as follows:

SMFID_PAT	RESOURCE_TYPE_PAT	OBJECT_NAME_PAT	cid_PAT	priority	enabled
%	SUBSYSTEM	%	MADR	10	1
%	WTO%	%	NULL	1	1

The second row specifies that any event with a RESOURCE_TYPE field that begins 'WTO' and which can be followed by any number of any characters should be discarded. Events are compared to this record before the other row because its priority column is set to 1, while the priority of the row with RESOURCE_TYPE_PAT='SUBSYSTEM' is 0.

The first row matches any event where the RESOURCE_TYPE is exactly equal to 'SUBSYSTEM' and the OBJECT_NAME is any value. Records that match this row are looked up as objects of class id MADR (MVS Address Space). The MVSAddressSpace object class is used as a default placeholder for discovered resources that cannot be classified according to any other row with a higher priority. The currently supported values for the *cid* column are listed in the following table.

Supported Resource Type Class Ids

cid	Description
CICS	CICS Region
DB2	DB2 Database Subsystem
IDMS	IDMS Database Subsystem
IMS	IMS Database Subsystem
MADR	MVS Address Space
STC	Started Task
OMEG	Omegamon Performance Monitor

As events reported by SA/390 are entered into the Tivoli Business Systems Manager database, they are classified by looking for the matching patterns in the *SA390ResourceTypeMap* table. The comparisons are done in descending priority order, so if a given event matches two or more entries in *SA390ResourceTypeMap*, the entry with the highest priority is considered the match. In the case where an event matches two or more entries in *SA390ResourceTypeMap* that have the same priority, the order of the match chosen is not guaranteed.

Once a match is found, the event is reported on an object of the matching class and name. If no object with the specified name exists, one is automatically created.

Configuring the SA390ResourceTypeMap Table

To customize the *SA390ResourceTypeMap* table according to a set of conventions based upon resource types and names, appropriate SQL (Standard Query Language) commands must be applied to the database. This has been facilitated by the stored procedure *DefineSA390ResourceTypeMapping*. This procedure has the following format:

DefineSA390ResourceTypeMapping *SMFID_PAT*, *RESOURCE_TYPE_PAT*, *OBJECT_NAME_PAT*, *cid*, *priority*, *enabled*

Examples are:

```
EXEC DefineSA390ResourceTypeMapping '%', '%', 'CIC%', 'CICS', 100, 1
EXEC DefineSA390ResourceTypeMapping '%', '%', 'DBB%', 'DB2', 100, 1
```

The above patterns declare that on any system any resource with a name that begins ‘CIC’ is to be classified as a CICS region, and that any resource that begins with the name ‘DBB’ is to be classified as a ‘DB2’ subsystem. The ‘%’ is a wildcard indicator that signifies any number of any characters. Single character wildcards are specified with the ‘_’ character, while ranges of matching characters can be specified by using the square-brackets ‘[’ and ‘]’ to define the range, (e.g. ‘[A-Z0-9]’ matches any single alphanumeric character). Negation is supported in the square brackets by using the ‘^’ character immediately after the opening ‘[’. For example, ‘[^0-9]’ specifies any non-numeric character.

To define your own mappings, create a text file modeled after the above examples, and apply this file to the database with a command of the following form (run from the command prompt):

```
isql -User -Ppassword -Sserver -iinput-file
```

where *user*, *password*, and *server* specify the database login parameters and *input-file* specifies the name of the file with your calls to ‘EXEC DefineSA390ResourceTypeMapping.’

You can test the currently defined mappings by using the stored procedure *TestSA390ResourceTypeMappings*. The format of this procedure is:

```
TestSA390ResourceTypeMapping SMFID, RESOURCE_TYPE, OBJECT_NAME
```

Invoking this procedure returns a list of all matching rows from *SA390ResourceTypeMap* for the specified *SMFID*, *RESOURCE_TYPE* and *OBJECT_NAME* strings.

Configuring State Change Message Format

Resources managed by automation receive state change messages that specify the new state of the resource. The event that reports the state change contains no event text per se, so Tivoli Business Systems Manager allows the system administrator to configure the format of the event text that is reported when these events occur. There are three fields reported in these events, which can be assembled into any form of event text that is desired. The three fields are:

RESOURCE_TYPE

type of resource managed by the automation system.

OBJECT_NAME

Name of the specific resource

STATE

New state of the resource

By default, when state change messages are posted to the objects in Tivoli Business Systems Manager they are posted with event text in the following format:

```
"RESOURCE_TYPE OBJECT_NAME IS IN STATE STATE"
```

where *RESOURCE_TYPE*, *OBJECT_NAME*, and *STATE* are the field values specified in the event.

This format can be changed by issuing the following SQL command:

```
master..sp_safermessage 50701, new-format-string [, @lang=language-specifier]
```

new-format-string is a quoted character string that must contain three occurrences of the string '%s', which is substituted with the appropriate field when the event is generated. The first occurrence of '%s' is substituted with *RESOURCE_TYPE*, the second with *OBJECT_NAME*, and the third with *STATE*.

If a language other than 'us_english' is specified by the @lang parameter, then the format string should contain the character sequences '%1!', '%2!', and '%3!' instead of '%s'. These may occur in any order and allow the event text to be configured appropriately to a specified language.

For more information on specifying messages, see the MSSQL Server documentation provided with your system.



Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.