

# **TIBCO Service Gateway™ for IMS/DB**

## **Installing and Operating**

*Software Release 6.0  
July 2012*

## Important Information

SOME TIBCO SOFTWARE EMBEDS OR BUNDLES OTHER TIBCO SOFTWARE. USE OF SUCH EMBEDDED OR BUNDLED TIBCO SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED TIBCO SOFTWARE. THE EMBEDDED OR BUNDLED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER TIBCO SOFTWARE OR FOR ANY OTHER PURPOSE.

USE OF TIBCO SOFTWARE AND THIS DOCUMENT IS SUBJECT TO THE TERMS AND CONDITIONS OF A LICENSE AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED SOFTWARE LICENSE AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER LICENSE AGREEMENT WHICH IS DISPLAYED DURING DOWNLOAD OR INSTALLATION OF THE SOFTWARE (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH SOFTWARE LICENSE AGREEMENT OR CLICKWRAP END USER LICENSE AGREEMENT, THE LICENSE(S) LOCATED IN THE "LICENSE" FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

This document contains confidential information that is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of TIBCO Software Inc.

TIBCO, The Power of Now, TIBCO Object Service Broker, and and TIBCO Service Gateway are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only.

THIS SOFTWARE MAY BE AVAILABLE ON MULTIPLE OPERATING SYSTEMS. HOWEVER, NOT ALL OPERATING SYSTEM PLATFORMS FOR A SPECIFIC SOFTWARE VERSION ARE RELEASED AT THE SAME TIME. SEE THE README FILE FOR THE AVAILABILITY OF THIS SOFTWARE VERSION ON A SPECIFIC OPERATING SYSTEM PLATFORM.

THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. TIBCO SOFTWARE INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "READ ME" FILES.

The TIBCO Object Service Broker technologies described herein are protected under the following patent numbers:

|            |         |         |         |         |         |          |
|------------|---------|---------|---------|---------|---------|----------|
| Australia: | -       | -       | 671137  | 671138  | 673682  | 646408   |
| Canada:    | 2284250 | -       | -       | 2284245 | 2284248 | 2066724  |
| Europe:    | -       | -       | 0588446 | 0588445 | 0588447 | 0489861  |
| Japan:     | -       | -       | -       | -       | -       | 2-513420 |
| USA:       | 5584026 | 5586329 | 5586330 | 5594899 | 5596752 | 5682535  |

Copyright © 1999-2012 TIBCO Software Inc. ALL RIGHTS RESERVED.

TIBCO Software Inc. Confidential Information

# Contents

|  |            |
|--|------------|
| <b>Preface</b> .....   | <b>vii</b> |
| Related Documentation .....                                  | viii       |
| TIBCO Object Service Broker Documentation .....              | viii       |
| Typographical Conventions .....                              | xiii       |
| Connecting with TIBCO Resources .....                        | xvi        |
| How to Join TIBCOCommunity .....                             | xvi        |
| How to Access All TIBCO Documentation .....                  | xvi        |
| How to Contact TIBCO Support .....                           | xvi        |
| <b>Chapter 1 Installing Service Gateway for IMS/DB</b> ..... | <b>1</b>   |
| Introduction .....   | 2          |
| Accessing IMS Data .....                                     | 3          |
| Processing IMS Data .....                                    | 4          |
| Deployment .....   | 5          |
| Preparing for Installation .....                             | 6          |
| Distribution Media and Contents .....                        | 7          |
| Obtaining the Installation Media .....                       | 7          |
| Installation Files .....                                     | 7          |
| Uploading the Software .....                                 | 8          |
| Installing the Software .....                                | 9          |
| Edit the Properties File .....                               | 9          |
| Initial Installation .....                                   | 10         |
| Installing on a Remote Host .....                            | 12         |
| Distribution Media and Contents .....                        | 12         |
| Uploading the Software .....                                 | 12         |
| Installing the Software .....                                | 14         |
| Uninstalling on a Remote Host .....                          | 21         |
| Installation Verification .....                              | 22         |
| Preparing the IMS Environment .....                          | 26         |
| Defining Gateway Program Specification Blocks .....          | 26         |
| Defining IMS Secondary Index Access .....                    | 27         |
| Authorizing Data Sets For Cross Memory Services .....        | 28         |
| Defining the Gateway to the IMS Online System .....          | 28         |
| Rollback Considerations For the DL/I Interface .....         | 28         |
| Preparing the DB2 Environment .....                          | 29         |

|  |           |
|--|-----------|
| Customized DB2 Plan Names .....  | 29        |
| Linking the Static SQL Handlers .....  | 29        |
| Preparing the CICS Environment .....   | 30        |
| Perform APPLY of the CICS DL/I Interface .....   | 30        |
| Customize the Program Control Table .....  | 30        |
| Modify the CICS Processing Program Table .....   | 30        |
| Add DDnames Used by the TIBCO Service Gateway for IMS/DB .....   | 30        |
| Implementing Security .....  | 31        |
| Specifying TIBCO Object Service Broker Security .....  | 31        |
| Implementing IMS Security .....  | 31        |
| Specifying the External Security Interface for DB2 Data Access .....   | 32        |
| DB2 Security for the Gateway .....   | 33        |
| Specifying Gateway Parameters .....  | 34        |
| Gateway Parameters .....   | 34        |
| Understanding the Startup Prerequisites .....  | 37        |
| Default Resource Settings (z/OS only) .....  | 37        |
| Starting the Gateway .....   | 39        |
| Starting in a Batch or Native Execution Environment Using the DL/I or BMP Interface .....  | 40        |
| Starting in a TIBCO Object Service Broker CICS Execution Environment Using the CICS Interface .....  | 41        |
| Starting in a Batch or Native Environment Using the DRA Interface .....  | 42        |
| Starting to Access Both IMS and DB2 Data in a TIBCO Object Service Broker Batch or Native Execution Environment Using the DL/I Interface ..... | 42        |
| Starting to Access Both IMS and DB2 Data in a TIBCO Object Service Broker Batch or Native Execution Environment Using the BMP Interface .....  | 44        |
| Dynamically Modifying Gateway Tasks .....  | 45        |
| Dynamically Starting Gateway Tasks .....   | 45        |
| Dynamically Setting Gateway Tasks .....  | 45        |
| Shutting Down the Gateway .....  | 47        |
| MODIFY Operator Command for Data Object Broker .....   | 47        |
| MODIFY Operator Command for the Execution Environment .....  | 48        |
| RESOURCE MANAGEMENT Option .....   | 48        |
| <b>Chapter 2 Operating the Service Gateway for IMS/DB .....</b>  | <b>49</b> |
| Defining IMS Databases to TIBCO Object Service Broker .....  | 50        |
| Completing the Definition for a Logical Database .....   | 53        |
| Binding IMS Table Definitions .....  | 55        |
| Preparing the IMS and DB2 Environments for Accessing DB2 Data .....  | 56        |
| Supplying Gateway Startup Parameters .....   | 57        |
| HRNIN Parameters .....   | 57        |
| IMSSRV00 Parameters .....  | 59        |
| Estimating the CTABLESIZE Parameter .....  | 72        |
| Dynamically Changing Gateway Parameters .....  | 74        |

|   |           |
|---|-----------|
| Table Type Attributes .....   | 74        |
| Using SETXPARM and RESETXPARM .....                                       | 74        |
| Gateway Parameters That Can Be Overridden at Runtime .....                | 75        |
| Examples Using SETXPARM and RESETXPARM .....                              | 75        |
| Implementing External Security .....                                      | 76        |
| Security Levels .....   | 76        |
| Authorizing the Gateway for SAF .....                                     | 77        |
| Establishing Resource Security .....                                      | 77        |
| Establishing PSB Security .....   | 78        |
| Establishing Database Security .....                                      | 79        |
| Establishing Segment Security .....                                       | 80        |
| Establishing Resource and Database Security .....                         | 80        |
| Establishing Resource and Segment Security .....                          | 81        |
| Establishing the TIBCO Object Service Broker Authorization ID .....       | 82        |
| Implementing Fail Safe Processing .....                                   | 84        |
| Procedural Overview .....   | 84        |
| Understanding Other Operational Procedures .....                          | 86        |
| Using Distributed Data with Service Gateway for IMS/DB .....              | 86        |
| Displaying the Status of the Gateway .....                                | 86        |
| Debugging Information .....   | 90        |
| Reporting Problems .....  | 91        |
| Connecting the Gateway to a Windows or Solaris Data Object Broker .....   | 92        |
| Configure the TCP/IP Connection on z/OS .....                             | 93        |
| Configure the TIBCO Object Service Broker TCP/IP Environment .....        | 93        |
| Specify the Number of Gateways Connecting to the Data Object Broker ..... | 94        |
| Specify the Gateway Parameter .....                                       | 94        |
| <b>Chapter 3 Managing IMS Data Definitions .....</b>                      | <b>95</b> |
| Managing IMS Data from TIBCO Object Service Broker .....                  | 96        |
| Task A: Extract the IMS Database Definition .....                         | 96        |
| Task B: Invoke the Table Definer .....                                    | 97        |
| Accessing Existing Tables .....   | 97        |
| Defining a New Table .....  | 97        |
| IMS Table Definition Screen Segments .....                                | 98        |
| PF Keys and Primary Commands .....  | 99        |
| Task C: Specify Header Information .....                                  | 101       |
| Table Definition Screen Fields .....                                      | 101       |
| Task D: Select an IMS Database .....                                      | 102       |
| Task E: Define a Location Parameter .....                                 | 106       |
| Minimal and Full Node Definitions .....                                   | 106       |
| Task F: Select IMS Segments .....   | 107       |
| Selecting Additional Segments .....                                       | 107       |

|  |            |
|--|------------|
| Task G: Select IMS Fields . . . . .                                  | 109        |
| Interpreting the Header Section . . . . .                            | 110        |
| IMS Field Section . . . . .  | 111        |
| Metadata Field Section . . . . .                                     | 112        |
| Task H: Define Event Rule Information . . . . .                      | 117        |
| Defining Fields . . . . .  | 117        |
| Defining Secondary Indexes . . . . .                                 | 119        |
| Defining Access Values . . . . .                                     | 119        |
| Defining Multiple Record Formats . . . . .                           | 120        |
| Mapping Data Types . . . . .   | 121        |
| Mapping Considerations . . . . .                                     | 123        |
| Effects of Segment Selection . . . . .                               | 123        |
| Retrieving and Updating Non-unique IMS Data . . . . .                | 123        |
| Effects of Field Selection . . . . .                                 | 124        |
| <b>Chapter 4 Processing IMS Data . . . . .</b>                       | <b>127</b> |
| Accessing TIBCO Object Service Broker IMS Tables . . . . .           | 128        |
| Using the Table Browser and Table Editor . . . . .                   | 128        |
| Browsing IMS Tables . . . . .  | 130        |
| Using Rules . . . . .  | 131        |
| Transaction Processing . . . . .                                     | 131        |
| Retrieval Processing . . . . .                                       | 132        |
| Replace (Update) Processing . . . . .                                | 134        |
| Insert Processing . . . . .  | 135        |
| Delete Processing . . . . .  | 135        |
| Understanding Gateway Behavior . . . . .                             | 136        |
| Synchronization and Recovery . . . . .                               | 136        |
| Gateway for IMS Exceptions . . . . .                                 | 137        |
| Using @SERVERERROR . . . . .   | 139        |
| <b>Appendix A Documenting IMS Tables . . . . .</b>                   | <b>141</b> |
| Using the Documentation Screen . . . . .                             | 142        |
| Field Values . . . . .   | 142        |
| PF Keys . . . . .  | 143        |
| <b>Appendix B Samples of Recommended Table Definitions . . . . .</b> | <b>145</b> |
| Sample Table Definitions . . . . .                                   | 146        |
| Sample IMS Database with Unique Segments . . . . .                   | 146        |
| Sample IMS Database with Non-unique Segments . . . . .               | 148        |
| <b>Index . . . . .</b>   | <b>151</b> |

# Preface



**This software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. Please see the readme file for the availability of this software version on a specific operating system platform.**

TIBCO® Object Service Broker is an application development environment and integration broker that bridges legacy and non-legacy applications and data. You can use TIBCO Object Service Broker to access external IMS and DB2 data and define TIBCO Object Service Broker tables based on this data. This manual describes the TIBCO Object Service Broker interface to IMS and concurrent access to DB2 data. The interface is known as the TIBCO Service Gateway for IMS/DB.

## Topics

---

- [Related Documentation, page viii](#)
- [Typographical Conventions, page xiii](#)
- [Connecting with TIBCO Resources, page xvi](#)

## Related Documentation

---

This section lists documentation resources you may find useful.

### TIBCO Object Service Broker Documentation

The following documents form the TIBCO Object Service Broker documentation set:

#### Fundamental Information

The following manuals provide fundamental information about TIBCO Object Service Broker:

- *TIBCO Object Service Broker Getting Started* Provides the basic concepts and principles of TIBCO Object Service Broker and introduces its components and capabilities. It also describes how to use the default developer's workbench and includes a basic tutorial of how to build an application using the product. A product glossary is also included in the manual.
- *TIBCO Object Service Broker Messages with Identifiers* Provides a listing of the TIBCO Object Service Broker messages that are issued with alphanumeric identifiers. The description of each message includes the source and explanation of the message and recommended action to take.
- *TIBCO Object Service Broker Messages without Identifiers* Provides a listing of the TIBCO Object Service Broker messages that are issued without a message identifier. These messages use the percent symbol (%) or the number symbol (#) to represent such variable information as a rules name or the number of occurrences in a table. The description of each message includes the source and explanation of the message and recommended action to take.
- *TIBCO Object Service Broker Quick Reference* Presents summary information for use in the TIBCO Object Service Broker application development environment.
- *TIBCO Object Service Broker Shareable Tools* Lists and describes the TIBCO Object Service Broker shareable tools. Shareable tools are programs supplied with TIBCO Object Service Broker that facilitate rules language programming and application development.
- *TIBCO Object Service Broker Release Notes* Read the release notes for a list of new and changed features. This document also contains lists of known issues and closed issues for this release.



## Application Development and Management

The following manuals provide information about application development and management:

- *TIBCO Object Service Broker Application Administration* Provides information required to administer the TIBCO Object Service Broker application development environment. It describes how to use the administrator's workbench, set up the development environment, and optimize access to the database. It also describes how to manage the Pagestore, which is the native TIBCO Object Service Broker data store.
- *TIBCO Object Service Broker Managing Data* Describes how to define, manipulate, and manage data required for a TIBCO Object Service Broker application.
- *TIBCO Object Service Broker Managing External Data* Describes the TIBCO Object Service Broker interface to external files (not data in external databases) and describes how to define TIBCO Object Service Broker tables based on these files and how to access their data.
- *TIBCO Object Service Broker National Language Support* Provides information about implementing the National Language Support in a TIBCO Object Service Broker environment.
- *TIBCO Object Service Broker Object Integration Gateway* Provides information about installing and using the Object Integration Gateway which is the interface for TIBCO Object Service Broker to XML, J2EE, .NET and COM.
- *TIBCO Object Service Broker for Open Systems External Environments* Provides information on interfacing TIBCO Object Service Broker with the Windows and Solaris environments. It includes how to use SDK (C/C++) and SDK (Java) to access TIBCO Object Service Broker data, how to interface to TIBCO Enterprise Messaging Service (EMS), how to use the TIBCO Service Gateway for WMQ, how to use the Adapter for JDBC-ODBC, and how to access programs written in external programming languages from within TIBCO Object Service Broker.
- *TIBCO Object Service Broker for z/OS External Environments* Provides information on interfacing TIBCO Object Service Broker to various external environments within a TIBCO Object Service Broker z/OS environment. It also includes information on how to access TIBCO Object Service Broker from different terminal managers, how to write programs in external programming languages to access TIBCO Object Service Broker data, how to interface to TIBCO Enterprise Messaging Service (EMS), how to use the TIBCO Service Gateway for WMQ, and how to access programs written in external programming languages from within TIBCO Object Service Broker.

- *TIBCO Object Service Broker Parameters* Lists the TIBCO Object Service Broker Execution Environment and Data Object Broker parameters and describes their usage.
- *TIBCO Object Service Broker Programming in Rules* Explains how to use the TIBCO Object Service Broker rules language to create and modify application code. The rules language is the programming language used to access the TIBCO Object Service Broker database and create applications. The manual also explains how to edit, execute, and debug rules.
- *TIBCO Object Service Broker Managing Deployment* Describes how to submit, maintain, and manage promotion requests in the TIBCO Object Service Broker application development environment.
- *TIBCO Object Service Broker Defining Reports* Explains how to create both simple and complex reports using the reporting tools provided with TIBCO Object Service Broker. It explains how to create reports with simple features using the Report Generator and how to create reports with more complex features using the Report Definer.
- *TIBCO Object Service Broker Managing Security* Describes how to set up, use, and administer the security required for an TIBCO Object Service Broker application development environment.
- *TIBCO Object Service Broker Defining Screens and Menus* Provides the basic information to define screens, screen tables, and menus using TIBCO Object Service Broker facilities.
- *TIBCO Service Gateway for Files SDK* Describes how to use the SDK provided with the TIBCO Service Gateway for Files to create applications to access Adabas, CA Datacom, and VSAM LDS data.

## System Administration on the z/OS Platform

The following manuals describe system administration on the z/OS platform:

- *TIBCO Object Service Broker for z/OS Installing and Operating* Describes how to install, migrate, update, maintain, and operate TIBCO Object Service Broker in a z/OS environment. It also describes the Execution Environment and Data Object Broker parameters used by TIBCO Object Service Broker.
- *TIBCO Object Service Broker for z/OS Managing Backup and Recovery* Explains the backup and recovery features of OSB for z/OS. It describes the key components of TIBCO Object Service Broker systems and describes how you can back up your data and recover from errors. You can use this information, along with assistance from TIBCO Support, to develop the best customized solution for your unique backup and recovery requirements.

- *TIBCO Object Service Broker for z/OS Monitoring Performance* Explains how to obtain and analyze performance statistics using TIBCO Object Service Broker tools and SMF records
- *TIBCO Object Service Broker for z/OS Utilities* Contains an alphabetically ordered listing of TIBCO Object Service Broker utilities for z/OS systems. These are TIBCO Object Service Broker administrator utilities that are typically run with JCL.

## System Administration on Open Systems

The following manuals describe system administration on open systems such as Windows or UNIX:

- *TIBCO Object Service Broker for Open Systems Installing and Operating* Describes how to install, migrate, update, maintain, and operate TIBCO Object Service Broker in Windows and Solaris environments.
- *TIBCO Object Service Broker for Open Systems Managing Backup and Recovery* Explains the backup and recovery features of TIBCO Object Service Broker for Open Systems. It describes the key components of a TIBCO Object Service Broker system and describes how to back up your data and recover from errors. Use this information to develop a customized solution for your unique backup and recovery requirements.
- *TIBCO Object Service Broker for Open Systems Utilities* Contains an alphabetically ordered listing of TIBCO Object Service Broker utilities for Windows and Solaris systems. These TIBCO Object Service Broker administrator utilities are typically executed from the command line.

## External Database Gateways

The following manuals describe external database gateways:

- *TIBCO Service Gateway for DB2 Installing and Operating* Describes the TIBCO Object Service Broker interface to DB2 data. Using this interface, you can access external DB2 data and define TIBCO Object Service Broker tables based on this data.
- *TIBCO Service Gateway for IDMS/DB Installing and Operating* Describes the TIBCO Object Service Broker interface to CA-IDMS data. Using this interface, you can access external CA-IDMS data and define TIBCO Object Service Broker tables based on this data.
- *TIBCO Service Gateway for IMS/DB Installing and Operating* Describes the TIBCO Object Service Broker interface to IMS/DB and DB2 data. Using this interface, you can access external IMS data and define TIBCO Object Service Broker tables based on it.

- *TIBCO Service Gateway for ODBC and for Oracle Installing and Operating*  
Describes the TIBCO Object Service Broker ODBC Gateway and the TIBCO Object Service Broker Oracle Gateway interfaces to external DBMS data. Using this interface, you can access external DBMS data and define TIBCO Object Service Broker tables based on this data.

## Typographical Conventions

The following typographical conventions are used in this manual.

Table 1 General Typographical Conventions

| Convention                           | Use   |
|--------------------------------------|---|
| <i>TIBCO_HOME</i><br><i>OSB_HOME</i> | <p>By default, all TIBCO products are installed into a folder referenced in the documentation as <i>TIBCO_HOME</i>.</p> <p>On open systems, TIBCO Object Service Broker installs by default into a directory within <i>TIBCO_HOME</i>. This directory is referenced in documentation as <i>OSB_HOME</i>. The default value of <i>OSB_HOME</i> depends on the operating system. For example on Windows systems, the default value is C:\tibco\OSB. Similarly, all TIBCO Service Gateways on open systems install by default into a directory in <i>TIBCO_HOME</i>. For example on Windows systems, the default value is C:\tibco\OSBgateways\6.0.</p> <p>On z/OS, no default installation directories exist.</p> |
| code font                            | <p>Code font identifies commands, code examples, filenames, pathnames, and output displayed in a command window. For example:</p> <p>Use MyCommand to start the foo process.</p>  |
| <b>bold code font</b>                | <p>Bold code font is used in the following ways:</p> <ul style="list-style-type: none"> <li>• In procedures, to indicate what a user types. For example: Type <b>admin</b>.</li> <li>• In large code samples, to indicate the parts of the sample that are of particular interest.</li> <li>• In command syntax, to indicate the default parameter for a command. For example, if no parameter is specified, MyCommand is enabled:<br/>MyCommand [<b>enable</b>   disable]</li> </ul>   |
| <i>italic font</i>                   | <p>Italic font is used in the following ways:</p> <ul style="list-style-type: none"> <li>• To indicate a document title. For example: See <i>TIBCO ActiveMatrix BusinessWorks Concepts</i>.</li> <li>• To introduce new terms. For example: A portal page may contain several portlets. <i>Portlets</i> are mini-applications that run in a portal.</li> <li>• To indicate a variable in a command or code syntax that you must replace. For example: MyCommand <i>PathName</i></li> </ul>  |

Table 1 General Typographical Conventions (Cont'd)




| Convention  | Use  |
|---|--|
| Key combinations  | <p>Key name separated by a plus sign indicate keys pressed simultaneously. For example: Ctrl+C.</p> <p>Key names separated by a comma and space indicate keys pressed one after the other. For example: Esc, Ctrl+Q.</p> |
|  | The note icon indicates information that is of special interest or importance, for example, an additional action required only in certain circumstances.   |
|  | The tip icon indicates an idea that could be useful, for example, a way to apply the information provided in the current section to achieve a specific result.   |
|  | The warning icon indicates the potential for a damaging situation, for example, data loss or corruption if certain steps are taken or not taken.   |

Table 2 Syntax Typographical Conventions

| Convention | Use  |
|------------|--|
| [ ]        | <p>An optional item in a command or code syntax.</p> <p>For example:</p> <p>MyCommand [optional_parameter] required_parameter</p>  |
|            | <p>A logical OR that separates multiple items of which only one may be chosen.</p> <p>For example, you can select only one of the following parameters:</p> <p>MyCommand para1   param2   param3</p> |

Table 2 Syntax Typographical Conventions

| Convention | Use   |
|------------|---|
| { }        | <p>A logical group of items in a command. Other syntax notations may appear within each logical group.</p> <p>For example, the following command requires two parameters, which can be either the pair param1 and param2, or the pair param3 and param4.</p> <pre>MyCommand {param1 param2}   {param3 param4}</pre> <p>In the next example, the command requires two parameters. The first parameter can be either param1 or param2 and the second can be either param3 or param4:</p> <pre>MyCommand {param1   param2} {param3   param4}</pre> <p>In the next example, the command can accept either two or three parameters. The first parameter must be param1. You can optionally include param2 as the second parameter. And the last parameter is either param3 or param4.</p> <pre>MyCommand param1 [param2] {param3   param4}</pre> |

## Connecting with TIBCO Resources

---

### How to Join TIBCOmmunity

TIBCOmmunity is an online destination for TIBCO customers, partners, and resident experts, a place to share and access the collective experience of the TIBCO community. TIBCOmmunity offers forums, blogs, and access to a variety of resources. To register, go to <http://www.tibcommunity.com>.

### How to Access All TIBCO Documentation

You can access TIBCO documentation here:

<http://docs.tibco.com>

### How to Contact TIBCO Support

For comments or problems with this manual or the software it addresses, please contact TIBCO Support as follows.

- For an overview of TIBCO Support, and information about getting started with TIBCO Support, visit this site:

<http://www.tibco.com/services/support>

- If you already have a valid maintenance or support contract, visit this site:

<https://support.tibco.com>

Entry to this site requires a user name and password. If you do not have a user name, you can request one.



## Chapter 1

# Installing Service Gateway for IMS/DB

This chapter introduces Service Gateway for IMS/DB and describes how to install, configure, and customize the software.

## Topics

---

- [Introduction, page 2](#)
- [Preparing for Installation, page 6](#)
- [Distribution Media and Contents, page 7](#)
- [Uploading the Software, page 8](#)
- [Installing the Software, page 9](#)
- [Installing on a Remote Host, page 12](#)
- [Installation Verification, page 22](#)
- [Preparing the IMS Environment, page 26](#)
- [Preparing the DB2 Environment, page 29](#)
- [Preparing the CICS Environment, page 30](#)
- [Implementing Security, page 31](#)
- [Specifying Gateway Parameters, page 34](#)
- [Understanding the Startup Prerequisites, page 37](#)
- [Starting the Gateway, page 39](#)
- [Dynamically Modifying Gateway Tasks, page 45](#)
- [Shutting Down the Gateway, page 47](#)

## Introduction

---

Service Gateway for IMS/DB is a server interface that provides concurrent, real-time access to IMS data from TIBCO Object Service Broker. It ensures that data is presented to TIBCO Object Service Broker rules in a manner consistent with TIBCO Object Service Broker behavior. The navigation of the IMS database is defined in the IMS table definition and is therefore not required in each application that accesses IMS data.

IMS data can be accessed through one of the following interfaces:

- DL/I
- BMP
- DRA
- CICS (IMS DB/DC, IMS DBCTL, Local DL/I, or Remote DL/I)

The DRA interface provides a multi-user, single address space environment for TIBCO Object Service Broker access to IMS data. In addition, you can access DB2 data concurrently through the IMS External Subsystem Attach Facility (ESAF). Your DB2 data must be defined to TIBCO Object Service Broker using the TIBCO Object Service Broker interface to DB2 data. Refer to [Chapter 2, Operating the Service Gateway for IMS/DB, on page 49](#) and *TIBCO Service Gateway for DB2 Installing and Operating* for more information.

You can also access IMS data concurrently through the IMS OTMA Callable Interface using the TIBCO Object Service Broker \$OTMA tool and the @OTMA\_MAP tool, which is a system interpreted table. Refer to *TIBCO Object Service Broker for z/OS External Environments* and *TIBCO Object Service Broker Shareable Tools* for information about this interface and these tools.

The Service Gateway for IMS/DB interface consists of the following components:

- **Table Definer** – Defines TIBCO Object Service Broker IMS tables based on the DBD information.
- **Service Gateway for IMS/DB or Service Gateway for IMS/DB2** – Formats TIBCO Object Service Broker retrieval and updates requests into DL/I calls.
- **Extract program** – Extracts IMS database definition (DBD) information and stores it in a sequential data set.
- TIBCO Object Service Broker **Load tool** – Loads the DBD information into control tables.

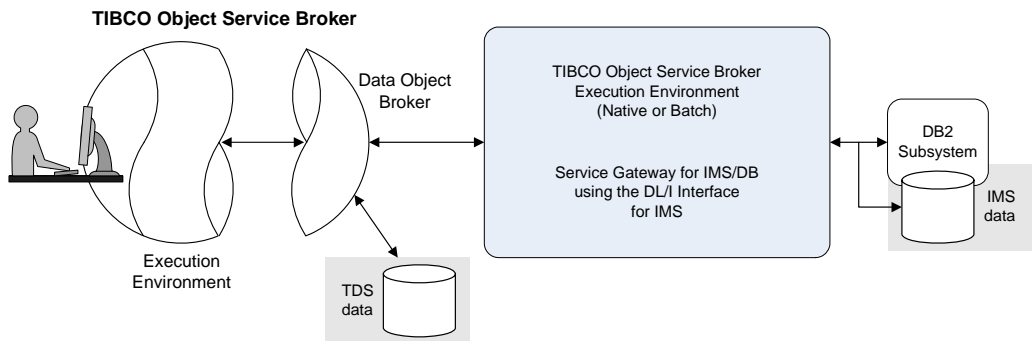
## Accessing IMS Data

TIBCO Object Service Broker uses IMS services to provide retrieval and update access, and to ensure data integrity. Service Gateway for IMS/DB executes within the following Execution Environments:

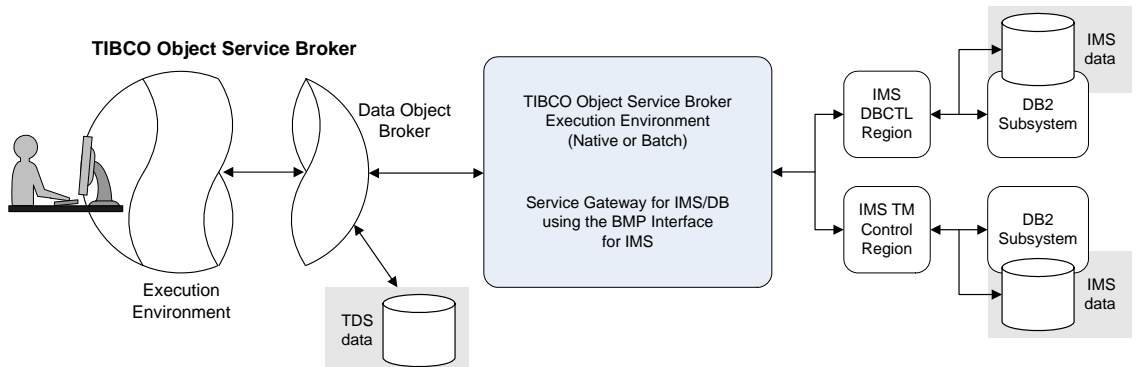
- TIBCO Object Service Broker Batch (DL/I, BMP, and DRA)
- TIBCO Object Service Broker Native Execution Environment (DL/I, BMP, and DRA)
- DRA
- CICS

The figures below show using these interfaces to access IMS data, while still having access to TDS data, TIBCO Object Service Broker's native data type.

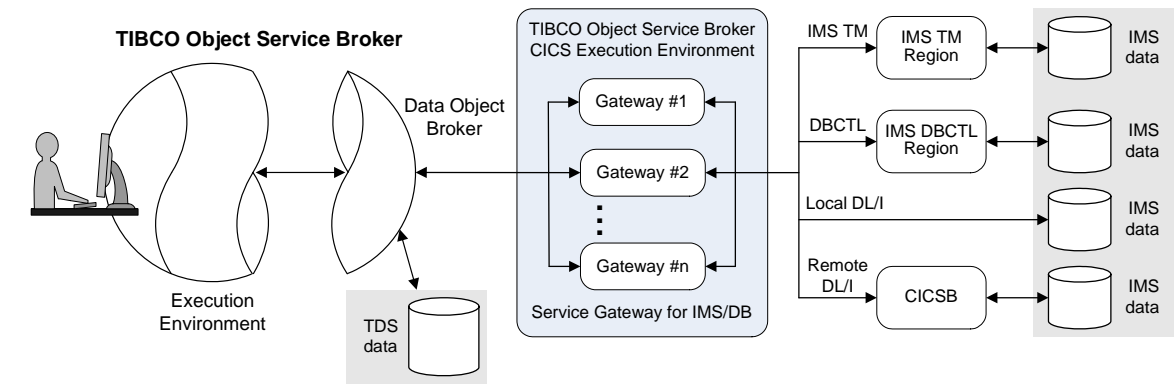
### Accessing IMS and DB2 Data Using the DL/I Interface for IMS



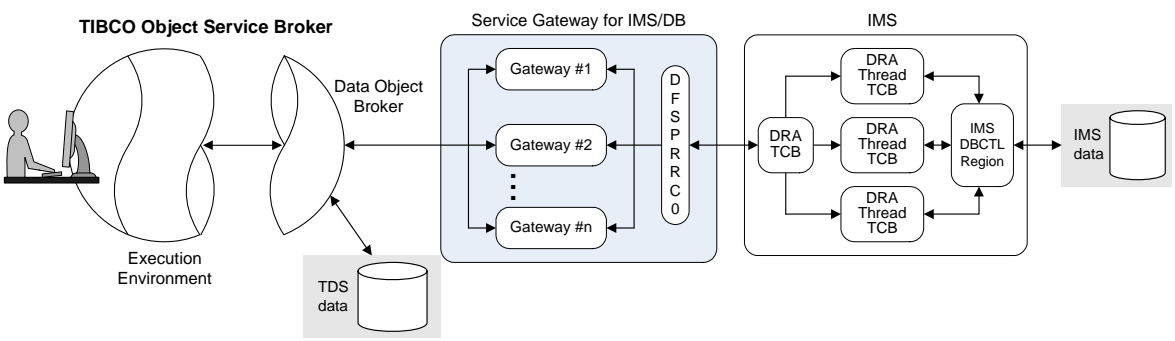
### Accessing IMS and DB2 Data Using the BMP Interface For IMS



Accessing IMS Data Using the CICS Interface For IMS



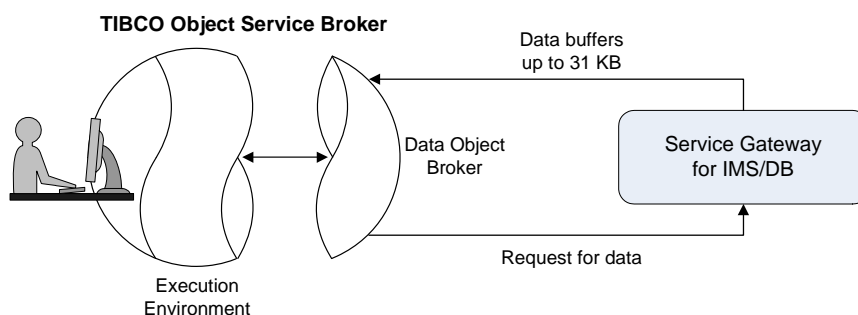
Accessing IMS Data Using the DRA Interface for IMS



Processing IMS Data

When a Gateway is started, it establishes communication with the Data Object Broker. TIBCO Object Service Broker is passed a SYSIN parameter file, which identifies the type of Gateway and the number of Gateway tasks to initialize. The SYSIN parameter file (identified by the IMSSRV00 DDname) specifies additional gateway parameters such as SERVERID and IDPREFIX.

When the Execution Environment requests access to IMS data, the appropriate DL/I call is built and sent to IMS TM or CICS. The illustration following shows how data is sent to the Data Object Broker in variable length buffers up to a maximum of 31 KB. If a single request requires more than 31 KB of data, multiple 31 KB buffers are sent until the request is complete.



## Deployment

You can configure the Data Object Broker and the Service Gateway for IMS/DB to reside on different hosts and/or operating systems (z/OS, Windows, or Solaris). Service Gateway for IMS/DB must be in the same domain as the IMS database system. See [Connecting the Gateway to a Windows or Solaris Data Object Broker on page 92](#) for additional information.



If all components reside in the same domain and in authorized libraries, Cross Memory Services is used for communications. Refer to [Chapter 2, Operating the Service Gateway for IMS/DB, on page 49](#) for information on running an authorized TIBCO Service Gateway for IMS/DB.

### See Also

*TIBCO Object Service Broker for z/OS Installing and Operating* or *TIBCO Object Service Broker for Open Systems Installing and Operating* for more information on communications requirements.

## Preparing for Installation

---



If you are installing this product on a remote host in relation to the Data Object Broker, where access to the product is only via a network, see [Installing on a Remote Host on page 12](#).

Before installing Service Gateway for IMS/DB, review the following:

- **TIBCO Object Service Broker Base Component** – You must install and ACCEPT (using SMP/E) the TIBCO Object Service Broker base component before installing Service Gateway for IMS/DB. You must also have the <HLQ>.INSTALL data set that was created during that installation.
- **Customizing the OSEMOD IMS and DB2 Variables** – Member OSEMOD in the CLIST data set is an ISPF edit macro used to customize members in the IMSDB.JOBS, CLIST, CNTL and JCL data sets. Refer to *TIBCO Object Service Broker for z/OS Installing and Operating* for the IMS and DB2 installation variables in OSEMOD that you should customize as required.
- **Fail Safe Processing** – To guarantee consistency when updating TDS, IMS, and DB2 data from a single instance of the Gateway in a single transaction, you must use Fail Safe level-1 processing. For details, see [Understanding Other Operational Procedures on page 86](#).

## Distribution Media and Contents

---

This section describes how to obtain the software, and the installation file that comprises the distribution media. Similar to the TIBCO Object Service Broker base component, the Service Gateway for IMS/DB software is distributed in .xm1 format within a zip file.

### Distribution File Format

The file is in a format compatible with IBM System Modification Program/Extended (SMP/E) naming conventions. The product is packaged in SMP/E txlib format.

## Obtaining the Installation Media

As with the TIBCO Object Service Broker base component, you can download the software from the TIBCO site by following these steps:

1. Contact TIBCO Software Inc. for a password, directory information, etc.
2. Connect to the TIBCO site with the required information.
3. Download the appropriate zip file.

## Installation Files

The following zip file comprises the distribution media:

TIB\_srvcgw-imsdb\_6.0\_zos.zip

## Uploading the Software

---

If you have acquired Service Gateway for IMS/DB by downloading it from the TIBCO Software web site, you must upload the software to the z/OS host system.

### Preparing and Uploading the Product File

1. Download or copy the `TIB_srvcgw-imsdb_6.0_zos.zip` file to a PC that can connect to the z/OS host system.
2. Unzip the file to a temporary location on the PC. The zip file contains multiple files; of these, the following file is the only file used in this installation:

`imsdb.xml` – compressed file containing Service Gateway for IMS/DB



The `srvcgw_imsdb.xml`, `install.bin`, `ostarrec.bin`, `property.bin`, and `OSTAREDC` files are not used in this procedure.

3. Pre-allocate the following sequential data set on the z/OS host system:

`<HLQ>.IMSDB.XM1` (size 17 KB)

Use the same `<HLQ>` that you specified when you uploaded the base component. Below is sample JCL to allocate this data set. Provide a JOB card and submit the JCL.

```
//ALLOC EXEC PGM=IEFBR14
//DD1 DD DSN=<HLQ>.IMSDB.XM1,
// DISP=(,CATLG,DELETE),UNIT=SYSDA,
// DCB=(RECFM=FB,LRECL=1024,BLKSIZE=0,DSORG=PS),
// SPACE=(TRK,(2,1))
```

4. FTP the `imsdb.xml` file in BIN mode to the `<HLQ>.IMSDB.XM1` data set.



## Installing the Software



You must perform the installation under an ISPF environment only.

This section describes the procedure for installing the Service Gateway for IMS/DB.

You can start the installation if you have the following data sets ready:

- <HLQ> . INSTALL
- <HLQ> . IMSDB . XM1



You must use the <HLQ> . INSTALL data set that was created during the installation of the TIBCO Object Service Broker base component.

### Installation Overview

To install Service Gateway for IMS/DB, perform the following:

1. Edit the properties file by specifying the keywords for installing this component.
2. Install the software.

## Edit the Properties File

Edit the PROPERTY member in <HLQ> . INSTALL. [Table 3](#) describes keywords in the properties file for installing this component.

*Table 3 Properties File Keywords*

| Keyword  | Description  |
|----------|--|
| INSTALL= | To install Service Gateway for IMS/DB, specify IMSDB:<br>INSTALL=IMSDB |
| RESLIB=  | Fully qualified IMS RESLIB library.                                    |

## Initial Installation

|                |   |
|----------------|---|
| <b>STEP 1:</b> | <b>Execute File Tailoring EXEC to start installation.</b>   |
| Member in:     | <HLQ>.INSTALL   |
| Member:        | INSTALL (EX member)   |
|                | The IMSDB.JCL data set is created at the successful completion of this step.  |
| <b>STEP 2:</b> | <b>Run Job IMSDB.JCL.</b>   |
|                | This batch job will uncompress the IMSDB.XM1 file to produce the distribution library.  |
| JCL in:        | <HLQ>.IMSDB.JCL (Edit JOB card to your site's standards)  |
| Data Set:      | <HLQ>.IMSDB.JCL (SUB data set)  |
|                | Uncompressing <HLQ>.IMSDB.XM1 produces the distribution library <HLQ>.IMSDB.FILEI.  |
| <b>STEP 3:</b> | <b>Create and customize work copies of data sets.</b>   |
| Member in:     | <HLQ>.IMSDB.FILEI   |
| Member:        | S6D1CUST (EX member)  |
|                | The following work copies are created and customized with values specified by OSEMOD variables:   |
|                | Customized copy - Library Description   |
|                | <ul style="list-style-type: none"> <li>• &lt;HLQNONV&gt;.&lt;INSTVER&gt;.JCL - Sample JCL</li> <li>• &lt;HLQNONV&gt;.&lt;INSTVER&gt;.IMSDB.JOBS - Install jobs for IMS/DB</li> </ul>  |
| <b>STEP 4:</b> | <b>Initiate install jobs.</b>   |
| Member in:     | <HLQNONV>.<INSTVER>.IMSDB.JOBS  |
| Member:        | S6D2RUNJ (EX member)  |
|                | SEND messages are directed to the userid specified in the NOTIFY parameter of each job submitted, informing user of submission and normal completion or abnormal termination. The successful completion of the final job in JOBSD list is accompanied by the message ALL MEMBERS PROCESSED. |
|                | This completes the installation process for IMS/DB.   |



You can modify the STATUS of any job as per your requirement. For example, if your shop normally ACCEPTs the product FMID at some future time, then change the status of S6D4ACPT from INSTALL to FUTURE. Note that you must ACCEPT the Service Gateway for IMS/DB component before applying any hotfix maintenance using SMP/E.

**STEP 5: Perform APPLY of CICS DL/I (Optional).**

If there is a requirement to run an instance of the Gateway in a CICS Execution Environment, modify the STATUS of job S6D5APLY in member JOBSD from OPTIONAL to INSTALL.

Member in: <HLQNONV> . <INSTVER> . IMSDB . JOBS  
 Member: JOBSD (EDIT member)  
 Member in: <HLQNONV> . <INSTVER> . IMSDB . JOBS  
 Member: S6D2RUNJ (EX member)

**STEP 6: Perform the final ACCEPT of CICS DL/I (Optional).**

If the standard in your site is to test a product prior to accepting it, leave the original STATUS of job S6D6ACPT as OPTIONAL so the job will not be submitted for execution. When you are ready to accept the Service Gateway for CICS DL/I product, perform the final ACCEPT.

Member in: <HLQNONV> . <INSTVER> . IMSDB . JOBS  
 Member: S6D6ACPT (SUB member)

## Installing on a Remote Host

---

This section describes the procedure for installing the software on a remote host in relation to the Data Object Broker installation.

### Distribution Media and Contents

This software is distributed in .xm1 format within a zip file. The file is in a format compatible with IBM System Modification Program/ Extended (SMP/E) naming conventions. The software is packaged in SMP/E txlib format.

#### Obtaining the Installation Media

You can download the software from the TIBCO site by following these steps:

1. Contact TIBCO Software Inc. for a password, directory information, etc.
2. Connect to the TIBCO site with the required information.
3. Download the appropriate zip file.

#### Installation Files

The following zip file comprises the distribution media:

`TIB_srvcgw-imsdb_6.0.0_zos.zip`

### Uploading the Software

If you have acquired the software by downloading it from the TIBCO site, you must upload the software to the z/OS host system.

#### Preparing and Uploading the Product File

1. Download or copy the `TIB_srvcgw-imsdb_6.0.0_zos.zip` file to a PC that can connect to the z/OS host system.

2. Unzip the file to a temporary location on the PC. The zip file contains multiple files; of these, the following files are the only files used in this installation:
  - `srvcgw_imsdb.xml` – compressed file containing Service Gateway for IMS/DB for installation on a remote host
  - `install.bin` – the REXX EXEC to perform the installation
  - `ostarrec.bin` – the REXX EXEC to uncompress the `.xml` file
  - `property.bin` – a template of mandatory install variables required for product installation.
  - `OSTAREDC` – a load module to improve the performance of `OSTARREC`



The `imsdb.xml` file is not used in this procedure.

3. Pre-allocate a PDS, fixed block data set on the z/OS host system with the following name:

```
<HLQ>.INSTALL
```

where `<HLQ>` is any valid high-level qualifier. Note that this `<HLQ>` will be used during the installation. See the sample JCL in the next step.

4. Pre-allocate the following sequential data set on the z/OS host system:

```
<HLQ>.OS.IMSDB.XM1 (size 46,220 KB)
```

Use the same `<HLQ>` as the previous data set. Below is sample JCL to allocate these data sets. Provide a JOB card and submit the JCL.

```
//ALLOC EXEC PGM=IEFBR14
//DD1 DD DSN=<HLQ>.INSTALL,
// DISP=(,CATLG,DELETE),UNIT=SYSDA,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=0),
// SPACE=(TRK,(5,15,100))
//DD2 DD DSN=<HLQ>.OS.IMSDB.XM1,
// DISP=(,CATLG,DELETE),UNIT=SYSDA,
// DCB=(RECFM=FB,LRECL=1024,BLKSIZE=0,DSORG=PS),
// SPACE=(TRK,(1000,50))
```

5. FTP `install.bin`, `property.bin` and `ostarrec.bin` to your z/OS system in BIN mode to the `<HLQ>.INSTALL` data set. Name these utilities `INSTALL`, `PROPERTY` and `OSTARREC`, respectively.
6. FTP the `srvcgw_imsdb.xml` file in BIN mode to the `<HLQ>.OS.IMSDB.XM1` data set.

## Installing the OSTAREDC Program

1. Upload the OSTAREDC file to z/OS in binary format to a data set with LRECL=80 and RECFM=FB.

2. In ISPF 3.4, against this data set, type the following:

```
"RECEIVE INDA(/)"
```

When prompted, specify DA( '<HLQ> . INSTLOAD' as the name of the load library where you want the OSTAREDC program restored, using the following syntax:

```
DA( 'datasetname' )
```

3. Edit OSTARREC as follows:

— Issue the command "FIND OSTAREDC 1".

— Change the constant after the equal sign to contain the full data set name of the program. The string must start with a double quote and a single quote, and end with a single quote and a double quote (the double quotes delimit the string and the single quotes tell TSO that the data set name is fully qualified). For example, change the following:

```
OSTAREDC = "'<HLQ> . INSTLOAD(OSTAREDC)'"
```

to

```
OSTAREDC = "'your.load.library(OSTAREDC)'"
```

where *your.load.library* is the name of the library referenced in Step 2.

## Installing the Software

You can start the installation if you have the following data sets ready:

- <HLQ> . INSTALL
- <HLQ> . OS . IMSDB . XM1



The <HLQ> referenced throughout this chapter is the high-level qualifier you specified when you uploaded the product software. This is the value of the INSTALL and XM1 files you specified. It will be used as the default value for all distribution files created when an XM1 is uncompressed. It is equivalent to the value of symbolic parameter \$HLQ\$ as described in OSEMOD.

## Installation Overview

To install Service Gateway for IMS/DB, perform the following:

1. Determine your system environment values listed in [System Environment Checklist](#).
2. [Edit the Properties File](#) using the values determined in Step 1.
3. [Install the Software](#).

## System Environment Checklist

Before you begin the installation, review the system environment information described in [Table 4](#) and determine whether you will use the default value or provide your own value.

Table 4 OSEMOD Variables

| Description  | OSEMOD Variable | Default Value          | Your Value |
|--|-----------------|------------------------|------------|
| High level qualifier for uploaded data sets<br>INSTALL and OS.IMSDB.XM1.                 | \$HLQ\$         | Specified on<br>upload |            |
| High level qualifier for non-VSAM and<br>VSAM data sets you are authorized to<br>create. | \$HLQNONV\$     | TIBCO.TESTNV           |            |
|  | \$HLQVSAM\$     | TIBCO.TESTVS           |            |
| Second level qualifier for install files.  | \$INSTVER\$     | INS60                  |            |
| Second level qualifier for TIBCO Service<br>Gateway system files.                        | \$SLQ\$         | OSB60                  |            |
| Second level qualifier for SMP/E libraries   | \$SMP\$         | SMP60                  |            |
| For SMS Shops – managementclass, dataclass and storageclass, if required                 |                 |                        |            |
| For new non-VSAM data sets.  | \$NMGTCLAS      | STANDARD               |            |
|  | \$NDATCLAS\$    | STANDARD               |            |
|  | \$NSTOCLAS\$    | S6BNONV                |            |
| For new VSAM data sets.  | \$VMGTCLAS      | STANDARD               |            |
|  | \$VDATCLAS\$    | STANDARD               |            |
|  | \$VSTOCLAS\$    | S6BVSAM                |            |

Table 4 OSEMOD Variables

| Description   | OSEMOD Variable | Default Value             | Your Value |
|---|-----------------|---------------------------|------------|
| High level qualifier of Language Environment libraries for SCEELKED and SCEEBIND. | \$CEELIB\$      | CEE                       |            |
| High level qualifier of IBM's Callable Services library CSSLIB.                   | \$CSSLIB\$      | SYS1                      |            |
| Fully qualified IMS RESLIB library.   | \$SDFSRESL\$    | IMS1010.IMS1.S<br>DFSRESL |            |

For additional information, refer to *TIBCO Object Service Broker for z/OS Installing and Operating*.

**Edit the Properties File**

Use the PROPERTY member in <HLQ> . INSTALL as a template, and modify to suit your requirements. [Table 5](#) describes keywords in the properties file that correspond to the system environment variables in [System Environment Checklist](#).

Table 5 Properties File Keywords

| Keyword         | Description  |
|-----------------|--|
| INSTALL=        | To install Service Gateway for IMS/DB, specify REMOTEGATEWAY:<br>INSTALL=REMOTEGATEWAY |
| SERVICEGATEWAY= | To install Service Gateway for IMS/DB, specify IMSDB:<br>SERVICEGATEWAY=IMSDB          |
| HLQNONV=        | High level qualifier for non-VSAM data sets.   |
| HLQVSAM=        | High level qualifier for VSAM data sets.   |
| INSTVER=        | Second level qualifier for install files.  |
| SLQ=            | Second level qualifier for TIBCO Object Service Broker system files.                   |
| SMP=            | Second level qualifier for SMP/E libraries.  |



Table 5 Properties File Keywords

| Keyword | Description   |
|---------|---|
| SMS=    | <p>YES for SMS site, NO for non-SMS site.</p> <p><b>Warning:</b> If you select the SMS=YES option, be sure to specify SMS-managed data-set names. The SMS automatic class selection (ACS) rules at your site determine whether a data-set name is eligible for SMS management. If the answer is yes, SMS manages that name. Otherwise, the result is unpredictable.</p>   |
| COMPAT= | <p>Use if SMS=YES . Valid values: YES for SMS compatible data set name classes; NO for SMS non-compatible data set name classes.</p> <p>If COMPAT=NO, specify the following:</p> <ul style="list-style-type: none"> <li>• NMGTCLAS – MANAGEMENTCLASS for non-VSAM data sets</li> <li>• NDATCLAS – DATACLASS for non-VSAM data sets</li> <li>• NSTOCLAS – STORAGECLASS for non-VSAM data sets</li> <li>• VMGTCLAS – MANAGEMENTCLASS for VSAM data sets</li> <li>• VDATCLAS – DATACLASS for VSAM data sets</li> <li>• VSTOCLAS – STORAGECLASS for VSAM data sets</li> </ul> |
| VOLSER= | <p>If SMS=YES, specify one DASD volume for VSAM data set allocation. Default is USER01. If SMS=NO, specify three DASD volumes separated by commas. Defaults are OSBS06, OSBD18, OSBB02.</p> <ul style="list-style-type: none"> <li>• vol1 – DASD volser for temp work files</li> <li>• vol2 – DASD volser for install files</li> <li>• vol3 – DASD volser for TIBCO Object Service Broker system files</li> </ul>   |
| CEELIB= | High level qualifier of Language Environment libraries.   |
| CSSLIB= | High level qualifier of IBM's Callable Services library CSSLIB.   |
| RESLIB= | Fully qualified IMS RESLIB library.   |

## Install the Software



To exit the interactive session at any time after executing the REXX exec INSTALL, do the following:

1. Press PA1
2. Enter hi
3. Press ENTER twice

### STEP 1: Execute File Tailoring EXEC to start installation.

Member in: <HLQ>.INSTALL

Member: INSTALL (EX member)

STEP 1 will verify that files can be allocated successfully using the values provided in the PROPERTY file. Test files of type sequential, PDS, PDSE, and VSAM will be allocated then deleted. Installation will stop if any test allocation fails. You should investigate the cause, correct the condition and repeat STEP 1.

The IMSDB.JCL data set is created at the successful completion of this step.

### STEP 2: Edit the Job card to your site's standards and run Job IMSDB.JCL.

JCL in: <HLQ>.IMSDB.JCL (Edit Job Card to your site's standards)

Data Set: <HLQ>.IMSDB.JCL (SUB data set)

This batch job will uncompress the OS.IMSDB.XM1 file to produce the distribution libraries. If you modify the job name, make sure it does not exceed seven characters. The job should successfully complete with a return code of 0.

### STEP 3: Edit OSEMOD. (Optional)

If you wish to make additional changes to the values of OSEMOD variables, make the changes now.

Member in: <HLQ>.FILECLS

Member: OSEMOD

**STEP 4: Create and customize work copies of data sets.**

Member in: <HLQ>.OS.IMSDB.FILEI

Member: S6Q1CUST (EX member)

The following work copies are created and customized with values specified by OSEMOD variables:

Customized copy – Library Description

- <HLQNONV>.<INSTVER>.CLIST – CLIST
- <HLQNONV>.<INSTVER>.CNTL – CNTL
- <HLQNONV>.<INSTVER>.JCL – Sample JCL
- <HLQNONV>.<INSTVER>.OS.IMSDB.JOBS – Install jobs for remote Service Gateway for IMS/DB

**STEP 5: Modify STATUS of installation jobs, as required.**

Member in: <HLQNONV>.<INSTVER>.OS.IMSDB.JOBS

Member: JOBSQ (EDIT member)

Jobs in MEMBER are evaluated in the order they are listed and are submitted based upon their specified STATUS. The next job is submitted only if the previous one completed with its expected return code RC.

Valid status: INSTALL (run the job), FUTURE/OPTIONAL (skip the job), DONE (job already completed).

Status is modified from INSTALL to DONE only if the job's completion code is equal to or less than the stated return code.

You can modify the STATUS of any job as per your requirement. For example, if your shop normally ACCEPTs the product FMID at some future time, then change the status of S6Q4ACPT from INSTALL to FUTURE. Note that you must ACCEPT the remote Service Gateway for IMS/DB FMID before applying any hotfix maintenance using SMP/E.

Skip this step if the default STATUS of all the jobs is acceptable.

**STEP 6: Initiate install jobs.**

Member in: <HLQNONV> . <INSTVER> . OS . IMSDB . JOBS

Member: S6Q2RUNJ (EX member)

SEND messages are directed to the userid specified in the NOTIFY parameter of each job submitted, informing user of submission and normal completion or abnormal termination. The successful completion of the final job in JOBSQ list is accompanied by the message ALL MEMBERS PROCESSED.

This completes the installation process.

**STEP 7: Perform APPLY of CICS DL/I (Optional).**

If there is a requirement to run an instance of the Service Gateway in a CICS Execution Environment, modify the STATUS of job S6Q5APLY in member JOBSQ from OPTIONAL to INSTALL.

Member in: <HLQNONV> . <INSTVER> . OS . IMSDB . JOBS

Member: JOBSQ (EDIT member)

Member in: <HLQNONV> . <INSTVER> . OS . IMSDB . JOBS

Member: S6Q2RUNJ (EX member)

**STEP 8: Perform the final ACCEPT of CICS DL/I (Optional).**

If the standard in your site is to test a product prior to accepting it, leave the original STATUS of job S6Q6ACPT as OPTIONAL so the job will not be submitted for execution. When you are ready to accept the Service Gateway for CICS DL/I product, perform the final ACCEPT.

Member in: <HLQNONV> . <INSTVER> . OS . IMSDB . JOBS

Member: S6Q6ACPT (SUB member)

## Uninstalling on a Remote Host

To uninstall the software, perform the following:

**STEP 1: Run the IMS/DB cleanup job.**

Member in: <HLQNONV>.<INSTVER>.OS.IMSDB.JOBS

Member: S6Q9CLEN (Edit JOBCARD and SUB member)

**STEP 2: Manually delete the following data sets (in the specified sequence):**

1. \$HLQ\$.FILECLS
2. \$HLQ\$.FILECTL
3. \$HLQ\$.FILEEEM1
4. \$HLQ\$.FILEEEM2
5. \$HLQ\$.FILEJCL
6. \$HLQ\$.FILEOBJ
7. \$HLQ\$.FILETRK
8. \$HLQ\$.FILEXML
9. \$HLQ\$.MACRO
10. \$HLQ\$.OS.IMSDB.FILEI
11. \$HLQNONV\$. \$INSTVER\$.CLIST
12. \$HLQNONV\$. \$INSTVER\$.CNTL
13. \$HLQNONV\$. \$INSTVER\$.JCL
14. \$HLQNONV\$. \$INSTVER\$.OS.IMSDB.JOBS

At this point, you should only have the uploaded data sets:

- \$HLQ\$.INSTALL
- \$HLQ\$.OS.IMSDB.XM1

Manually delete the data sets.

## Installation Verification

---

Installation verification for an external DBMS provides a quick method to verify that the installation of a TIBCO Object Service Broker DOB and one or more DBMS Service Gateways was successful. This verifies that the communication between the DOB and a Service Gateway, and a Service Gateway and the DBMS, is functioning properly.

The verification procedure is split between two elements. First, the IVPIMSP member that was placed in <HLQNONV> . <INSTVER> . JCL by the installation procedure for the Service Gateway for IMS/DB, creates a PSB with PROCOPT=GP, copied from DFSIVP6, and starts the Gateway.

Once the Service Gateway is running, the second element, IVPIMS, must be run in the environment of the Data Object Broker. IVPIMS is a batch file, shell script, or JCL member, which loads a DBD file corresponding to the IBM-supplied IMS table IVPDB1, and accesses data in the external database to verify connectivity.

Each of these steps can be done manually, meaning you can start the Service Gateway for IMS/DB normally rather than by using IVPIMSP, and you can browse data in the external database via the Gateway without using the shell script/batch file/JCL.

TIBCO Object Service Broker tables are pre-defined with definitions that correspond to the sample tables or demo databases commonly included with the various external DBMSs. In the case of DB2, the table IVPDB1 is used. The TIBCO Object Service Broker tables are prefixed with the name S6BIVP\*, for example S6BIVP\_IMS.



If a DBMS does not have sample tables or a demo database, or these were not included in its installation, you need to manually verify access to one of your databases. Instructions to perform this can be found later in this manual.

### Requirements

Installation verification requires the following:

- The complete installation of a TIBCO Object Service Broker DOB on z/OS, Windows or UNIX.
- The installation of a Service Gateway that must be run from an APF authorized library. If the DOB is running on z/OS, the Gateway must be properly configured in Resource Management.

- The sample table IVPDB1, which is commonly installed with IMS. The IVP depends upon the presence of data in this table, but does not rely on any specific values within the table.
- The IBM-supplied data sets IMS1010.IMS1.DFSIVD1 and IMS1010.IMS1.DFSIVD1I (or versions appropriate to your version of IMS/DB) must be present, and the user must have TSS/RACF permissions to access these data sets. Note that if your version of these files are different from these, the data set names in the IVP itself will have to be changed to match. See variables \$IMSIVPDB\$ and \$IMSIVPIX\$ in OSEMOD.
- The configuration of the communications path between the Service Gateway and the Data Object Broker.
- A valid level 1 TIBCO Object Service Broker user Id must exist matching the IDPREFIX parameter specified in the Gateway startup JCL.

Before you run the Installation Verification Procedure (IVP), you must configure the communication path that the Service Gateway will use to communicate with the Data Object Broker. The details of this configuration depend upon whether the Data Object Broker is installed on z/OS or on an open systems platform, and upon the communication protocol to be used.

- If the DOB is on z/OS and you are not using TCP/IP to communicate between it and the Gateway, comment out the DD statement for S6BRELAY in the IVPIMSP JCL member.
- If the DOB is on Windows or UNIX, and the communications protocol is TCP/IP, then the communications configuration for the Service Gateway is determined by the relay file placed in <HLQNONV>.<SLQ>.RELAYCFG by the installer. A complete discussion of how to configure the relay file is beyond the scope of this document; for details, see *TIBCO Object Service Broker for z/OS Installing and Operating*.

For the purpose of the IVP only, a simplified file will suffice. The following example illustrates a relay file that configures a connection to a remote DOB on Windows/Unix.

---

```
<relay xmlns="http://www.tibco.com/OSB/relayparms.xsd">
  <tcpipparms tcbnum="3" maxtcbsockets="50" />
  <directory>
    <node name="WINDOB">
      <tcpip host="192.168.1.1" port="12000" />
    </node>
  </directory>
</relay>
```

---

Substitute the name, host, and port for your DOB into this template. They can be found in the `HURON.DIR` file in the database directory of the TIBCO Object Service Broker installation to which you are trying to connect.

Member `IVPIMSP` in data set `<HLQNONV> . <INSTVER> . JCL` contains the job steps required to prepare the Service Gateway for the verification process. If all requirements are met, customize the JCL and run it. Note that if the Service Gateway has already been configured and started, this step is not necessary.

To customize the JCL, replace each instance of `$TDS$` in the JCL with the DOB node name that you placed in the relay file.

In the same JCL, note the value of `IDPREFIX`. A valid TIBCO Object Service Broker user Id with this name must exist in the DOB to which you will be connecting.

### IVP Batch File, Shell Script or JCL

The IVP is a batch file, shell script, or JCL member in the environment of the DOB.

- On Windows, the batch file `ivpims.bat` is in the `bin` directory of your TIBCO Object Service Broker installation. Variables within the file need to be customized; directions for doing so are in the file. The customization includes setting where the results will be placed.
- On Solaris, the (ksh) shell script `ivpims.ksh` is in the `bin` directory of your TIBCO Object Service Broker installation. Shell and environment variables within the script need to be customized; directions for doing so are in the script. The customization includes setting where the results will be placed.
- On z/OS the JCL member `IVPIMS` will be found in the `<HLQNONV> . <INSTVER> . JCL` data set associated with the DOB. The customization will involve supplying values for `TDS`, `USERID`, and `PASSWORD`. The results will be placed in the job's output, and may be viewed using `SDSF`.

### Verification Process

The steps in the verification process are described below.

Manually, or by using `IVPIMSP`:

1. Perform any DBMS preparation work, such as establishing a Program Specification Block.
2. Submit JCL to start the Service Gateway for IMS/DB.

At the DOB's node, using `ivpims.bat`, or the corresponding shell script or JCL member:



1. Run a TIBCO Object Service Broker batch job to access the predefined DBMS sample table.
2. Check the results. Sample data from the database will be present in the output if the verification procedure succeeded.

## Preparing the IMS Environment

---

Before running Service Gateway for IMS/DB, prepare the following items for the IMS environment:

- Define Program Specification Blocks (PSBs) for each class of Gateway.
- If using Cross Memory Services for the Gateway to communicate with the Data Object Broker, authorize specific data sets.
- If using the BMP, CICS DL/I, or DRA interface, define the Gateway PSB to the online IMS system.
- Ensure the Offline DL/I Batch Processing Region can handle dynamic backout.

You must also consider the level of security required for your environment and whether TIBCO Object Service Broker, IMS, and DB2 data must be updated in a single transaction. Refer to [Implementing Security on page 31](#) and [Implementing Fail Safe Processing on page 84](#) for more information.

### Defining Gateway Program Specification Blocks

Each class of the Gateway requires a PSB that defines all the IMS databases that you want to access from TIBCO Object Service Broker through that class of Gateway, and the processing options available for each database. You must define a database Program Control Block (PCB) for each database you need the Gateway to access. The PSB defined for the BMP, CICS DL/I and DRA interface is locked for the time of the TIBCO Object Service Broker transaction. The CICS interface can use different PSBs but only one per TIBCO Object Service Broker transaction.

If a database contains non-unique segments (no key or multiple sequence keys), define at least two identical PCBs. The first PCB is reserved for unique segment access. Subsequent PCBs are used to maintain the position in the database when accessing non-unique segment occurrences. The number of PCBs required is the maximum number of non-unique IMS tables that map to the same database, accessed in a single TIBCO Object Service Broker transaction.

#### Gateway PSB Requirements:

- The Gateway uses PATH calls, therefore, include the P processing option in the PROCOPT (Processing Option) parameter for each database PCB (for example, AP or IRDP). If you specify the PROCOPT (Processing Option) parameter on a SENSEG statement, include the P option, otherwise the SENSEG specification overrides the PCB specification. This could result in a DL/I AM status code failure.

- You must include the CMPAT=YES option on the PSBGEN statement because the Gateway expects the first PCB to be an I/O PCB.
- Include the LANG=ASSEM option.

The following is a sample PSB definition:

---

```
* Sample PSB definition suitable for TIBCO Object Service Broker
*
CUSTOMER PCB    TYPE=DB,DBDNAME=CUSTDB,PROCOPT=AP,KEYLEN=28
               SENSEG NAME=CUSTROOT,PARENT=0
               SENSEG NAME=CUSTDEF1,PARENT=CUSTROOT
               SENSEG NAME=ADDRESS,PARENT=CUSTROOT,PROCOPT=IRDP
*
*   The TIBCO Object Service Broker FSLEVEL=1 database
*   See: <HLQNONV>.<INSTVER>.CNTL(XIMSTRXP)
*
HRNTRXDDB PCB   TYPE=DB,NAME=HRNTRXDDB,PROCOPT=AP,KEYLEN=16
               SENSEG NAME=HRNTRXDDB,PARENT=0
*
               PSBGEN PSBNAME=CUSTPSB,LANG=ASSEM,CMPAT=YES
               END
```

---

## Defining IMS Secondary Index Access

If you always access an IMS database through a secondary index, include the PROCSEQ parameter with all PCBs defined for that database. If you access an IMS database using both primary and secondary processing sequences, complete the following steps:

1. Create an IMS logical database following IMS restrictions.  
If the physical database has no logical relationships, the sequence of segments is exactly the same as the physical database definition.
2. Define PCBs (which include the PROCSEQ parameter) for the logical database in the Gateway PSB.  
The sequence of segments should reflect the order in which segments are returned when using secondary index processing. However, if the INDICES parameter is defined (instead of PROCSEQ), the sequence of the segments stays the same as the logical database definition.
3. You can now map the IMS table definitions to different databases, depending on how you need to access the database.

For example, to retrieve data using the primary processing sequence, create an IMS table definition using the physical database definition.

To retrieve data using the secondary processing sequence (using a secondary index), create an IMS table definition using the logical database and specify the secondary index fields (Idx=Y) as parameters or an access value for the table.

## Authorizing Data Sets For Cross Memory Services

If an instance of the Gateway uses Cross Memory Services to communicate with the Data Object Broker, ensure that all data sets in the STEPLIB concatenation list are APF authorized.

## Defining the Gateway to the IMS Online System

To run the Gateway using the BMP interface to IMS DB/TM or DBCTL, define the IMS APPLCTN macro as shown:

```
APPLCTN  PGMTYPE=BATCH, PSB=HRNIMSS, SCHDTYP=PARALLEL
```



Include the SCHDTYP=PARALLEL portion so multiple Gateways can use the same PSB.

## Rollback Considerations For the DL/I Interface

The Gateway uses the DL/I ROLB call to roll back any updates if they are required or requested. In order for the ROLB call to be successful, if using the DL/I interface, you must set the dynamic backout parameter to Yes (BKO=Y). An IEFORDER data set is also required for IMS logging. Ensure that this data set remains on DASD and is large enough to accommodate rollback processing. Refer to [Starting the Gateway on page 39](#) for more information.

## Preparing the DB2 Environment

---

If you require concurrent access to DB2 data, the Static SQL Handlers must be linked.

### Customized DB2 Plan Names

Depending on which interface to IMS data your Gateway is using, the following occurs with respect to DB2 plan names:

- A Service Gateway for IMS/DB2 using the BMP interface uses the name specified in the PSBNAME parameter as the DB2 plan name, unless the PLAN (in IMSSRV00) or SERVERPARM (in HRNIN) parameter is used to specify a different plan name. The DB2 plan must be defined to DB2 as specified for the Service Gateway for IMS/DB2. Refer to [Supplying Gateway Startup Parameters on page 57](#) for more information.
- A Service Gateway for IMS/DB2 using the DL/I interface uses the DB2 plan name that is passed through the SYSIN DD statement (DDITV02).

### Linking the Static SQL Handlers

The Static SQL handlers must be properly linked. Add the default AMODE and RMODE to the link control card input statements: AMODE(31) and RMODE(24)

See Also *TIBCO Service Gateway for DB2 Installing and Operating* for more information on installing the Service Gateway for DB2 and linking the Static SQL handlers.

## Preparing the CICS Environment

---

To run an instance of the Gateway in a CICS Execution Environment, perform the tasks in this section. To access IMS data using the CICS interface, you must install the interface.

### Perform APPLY of the CICS DL/I Interface

For details on performing the APPLY of the CICS DL/I Interface, see [Installing the Software on page 9](#).

### Customize the Program Control Table

You can add HSRV as the TIBCO Object Service Broker CICS transaction identifier (trans ID) for the Gateway for to the Program Control Table (PCT) MRO or non-MRO. The CICS Execution Environment assigns this trans ID to all CICS tasks associated with the Gateway if it is specified to the CICSHURONTRAN parameter in the HRNIN SYSIN file. This entry can specify the same program name as the entry for the trans ID HURN. For more, refer to [Specifying Gateway Parameters on page 34](#).

### Modify the CICS Processing Program Table

In addition to the TIBCO Object Service Broker CICS Execution Environment interface programs defined in your CICS Processing Program Table (PPT), you must include an entry for program S6BCSDLI. For example:

```
DFHPPT TYPE=ENTRY, PGMLANG=ASSEMBLER, PROGRAM=S6BCSDLI, RES=YES
```

### Add DDnames Used by the TIBCO Service Gateway for IMS/DB

In addition to the DDnames used by TIBCO Object Service Broker under CICS, you must add DDname IMSSRV00 to the CICS Region startup JCL. The SYSIN member associated with this DDname contains the Gateway startup parameters. For more information, refer to [Starting the Gateway on page 39](#).

See Also *TIBCO Object Service Broker for z/OS Installing and Operating* for more information on installing the CICS interface. See *TIBCO Object Service Broker for z/OS External Environments* for more information on the DDnames used by TIBCO Object Service Broker under CICS.

## Implementing Security

---

The following types of security are available from the Gateway:

- TIBCO Object Service Broker security
- IMS security
- External security for DB2 data access
- DB2 security

### Specifying TIBCO Object Service Broker Security

To restrict the ability to define IMS tables, restrict read access to the @IMSTABLES and @IMSFIELDS TIBCO Object Service Broker control tables.

### Implementing IMS Security

The Gateway provides a number of methods of authorizing access to IMS resources. This is determined by the combination of the SECLEVEL parameter and various other gateway parameters. The table below explains each of the seven methods. Refer to [Supplying Gateway Startup Parameters on page 57](#) and [Implementing External Security on page 76](#) for more information on IMS security.

| Gateway Parameters Specified |                   | Explanation   |
|------------------------------|-------------------|---|
| SECLEVEL=0                   |                   | All IMS database accesses are verified using the ID that started the Gateway using the DL/I interface (if running as a batch job) or the started task name (if running as a started task). The Gateway using the BMP interface is authorized as specified by the existing IMS security. |
| SECLEVEL=1                   | CLASS<br>RESOURCE | The TIBCO Object Service Broker session ID, current security group, or a combination of both is used to verify access to the specified CLASS and RESOURCE.  |

| Gateway Parameters Specified |                               | Explanation  |
|------------------------------|-------------------------------|--|
| SECLEVEL=1                   | CLASS<br>RESOURCE<br>DBDCLASS | The TIBCO Object Service Broker session ID, current security group, or a combination of the two is used to verify access to the specified CLASS, RESOURCE, DBDCLASS, and IMS database.             |
| SECLEVEL=1                   | DBDCLASS                      | The TIBCO Object Service Broker session ID, current security group, or a combination of the two is used to verify access to the specified DBDCLASS and IMS database.                               |
| SECLEVEL=1                   | CLASS<br>RESOURCE<br>SEGCLASS | The TIBCO Object Service Broker session ID, current security group, or a combination of the two is used to verify access to the specified CLASS, RESOURCE, SEGCLASS, and IMS database and segment. |
| SECLEVEL=1                   | SEGCLASS                      | The TIBCO Object Service Broker session ID, current security group, or a combination of the two is used to verify access to the specified SEGCLASS, and IMS database and segment.                  |
| SECLEVEL=1                   | PSBCLASS                      | The TIBCO Object Service Broker session ID, current security group, or a combination of the two is used to verify access to the specified PSBCLASS and PSB.  |

Specifying the External Security Interface for DB2 Data Access

If the external security interface is requested (SECLEVEL=1) in the initialization parameters, only the IMS external security interface is invoked. The DB2 external security interface is not invoked. Refer to [Implementing External Security on page 76](#) and *TIBCO Service Gateway for DB2 Installing and Operating* for more information on the external security interface.



## DB2 Security for the Gateway

If the Gateway is using the DL/I or BMP interface, the primary authorization ID passed to DB2 is as follows:

| If ...   | The authorization ID passed to DB2 is ...   |
|--|---|
| Running as a batch job   | The user ID that submitted the Gateway job. |
| Running as a started task  | The started task name.                      |
| An external security package, such as RACF, CA-ACF2, or CA-Top Secret, is not available to build the z/OS field ASXBUSER (address space extension block) | The PSB name.                               |

See Also *TIBCO Object Service Broker Managing Security* for more information on restricting table access.

*DB2 Administration Guide* for information on controlling access to a DB2 subsystem.

*TIBCO Service Gateway for DB2 Installing and Operating* for information on the security required by the primary authorization ID.

# Specifying Gateway Parameters

You can specify gateway parameters in the HRNIN DD statement in the startup JCL or in the IMSSRV00 SYSIN file.



The PSBNAME and SERVERID parameters can be overridden at runtime. Refer to [Dynamically Changing Gateway Parameters on page 74](#) for more information.

## Gateway Parameters

The tables below show a list of all HRNIN parameters, all IMSSRV00 general, Fail Safe, and security parameters. Default values (if any) are also shown. These parameters are described fully in [Supplying Gateway Startup Parameters on page 57](#).

### HRNIN Gateway Parameters

| Parameter     | Default  | Required |
|---------------|--|----------|
| MDL           | OSB9999  |          |
| SERVERS       | 1  |          |
| SERVERTYPE    | None   | Y        |
| TDS           | None   | Y        |
| CICSHURONTRAN | None—applicable only to Gateways running in a TIBCO Object Service Broker CICS Execution Environment and using the CICS interface.   | Y        |
| SERVERPARM    | If access is only IMS data:<br>'DLI , S6BDRSSR , S6BIMSS , , , , , , , , , N , , , Y'<br>If access is both IMS and DB2 data:<br>'DLI , DSNMTV01 , S6BIMSS , , , , , , , , , N , , , Y' | Y        |
| DRASUFFIX     | Applicable only to Gateways using the DRA interface.   | Y        |

**IMSSRV00 General Gateway Parameters**

| Parameter       | Default  |
|-----------------|--|
| DEBUG           | Not Applicable. Include parameter or omit.   |
| DLIUPDATELIMIT  | 0  |
| DLIQUERYLIMIT   | 0  |
| IDPREFIX        | IMS01  |
| MODE            | UPD  |
| PLAN            | PSBNAME  |
| POOLSIZE        | 256  |
| PSBNAME         | S6BIMSS  |
| RESPONSEMODE    | ASYNC  |
| SCOPE           | TRANSACTION in a CICS environment.<br>SESSION in a non-CICS environment.   |
| SERVERID        | DEFAULT  |
| THREADUSAGE     | TRANSACTION – for IMS Gateways running in a CICS address space.<br>SESSION – for DRA servers.  |
| TRACE           | Not Applicable. Include parameter or omit.   |
| USERTYPE        | ANY – For Gateways with both batch and online users. This is the default for Gateways running in a Batch or Native Execution Environment.<br><br>ONLINE – For online Gateways only (BMP or CICS DL/I running under S6BDR000).<br><br>LOCAL – To execute the TIBCO Object Service Broker transaction in the same address space as the Gateway. Useful for CICS DL/I and Batch Gateways. |
| VERIFYEXTACCESS | CALL   |

IMSSRV00 Fail Safe Gateway Parameters

| Parameter   | Default  |
|-------------|--|
| FSLEVEL     | 0  |
| FSTABLENAME | @DB2FSTRXDB—Applicable only if concurrently accessing both IMS and DB2 data. |
| TRXDB       | None.  |

IMSSRV00 Security Gateway Parameters

| Parameter      | Default                       |
|----------------|-------------------------------|
| CLASS          | None.                         |
| DBDCLASS       | None.                         |
| DECOUPL        | NO                            |
| EXTERNALGROUP  | None.                         |
| EXTERNALUSERID | USERID                        |
| PSBCLASS       | None.                         |
| RECOVERYID     | ID used to start the Gateway. |
| REQSTOR        | S6BIMSS                       |
| RESOURCE       | None.                         |
| SECLEVEL       | 0                             |
| SEGCLASS       | none                          |
| SUBSYS         | HURON                         |

# Understanding the Startup Prerequisites

Before you can start an instance of the Gateway, you must do one of the following:

- If the Data Object Broker is on z/OS, the server must be identified to the Data Object Broker. To do this, define Gateway resources to the Data Object Broker’s resource management repository file.
- If the Data Object Broker is on Windows, set up National Language Support, if necessary. Refer to *TIBCO Object Service Broker National Language Support* for setup and configuration information.

## Default Resource Settings (z/OS only)

Use the Resource Management option (option 3) available from the Administration control group of the TIBCO Object Service Broker Administration Menu. You need to use the Resource Details (PF5) and the Resource Schedules (PF10) screens to specify the connection attributes. To get to the Resource Detail screen, you must first specify a type (SERVERTYPE) and group (SERVERID) on the Resource Type screen.

The following table illustrates the attributes for SERVERID=DEFAULT:

| Resource Details    |               |                 |              | Resource Schedule |
|---------------------|---------------|-----------------|--------------|-------------------|
| INTERMEDIATE ROLLBK | EARLY RELEASE | LAST USER REUSE | COMMIT LEVEL | ONLINE ONLY       |
| N                   | Y             | N               | 0            | N                 |



- If you are accessing IMS data using the CICS interface, the ONLINE ONLY default is Y. The Group ID is the server ID for the Gateway, or job name if USERTYPE=LOCAL.
- You can get the Resource Manager to dynamically assign a name that is a combination of the resource type and the group name. This is done using the wildcard facility of the Resource Manager.

Default Settings

The settings of the fields are affected by the values of the following gateway parameters:

- RESPONSEMODE
- FSLEVEL
- USERTYPE

Depending on the values you set in [Specifying Gateway Parameters on page 34](#), you must specify the following defaults:

| If Gateway Parameter ... | Then ...  |
|--------------------------|---|
| RESPONSEMODE=SYNC        | EARLY RELEASE must be set to N.   |
| RESPONSEMODE=ASYNC       | EARLY RELEASE must be set to Y.   |
| FSLEVEL=1                | COMMIT LEVEL must be set to 1.  |
| USERTYPE=ONLINE          | ONLINE ONLY must be set to Y.   |
| USERTYPE=ANY             | ONLINE ONLY must be set to N, since the default if using the CICS interface is Y. |
| USERTYPE=LOCAL           | ONLINE ONLY can be set to Y or N.   |

See Also *TIBCO Object Service Broker for z/OS Installing and Operating* for more information on defining and managing resources, and the Administration Menu.

## Starting the Gateway

---

When you start an instance of the Gateway, you are connected to TIBCO Object Service Broker. When the Execution Environment requests an access to IMS data, the Gateway establishes a connection to IMS (DL/I, BMP, DRA or CICS) and builds the corresponding DL/I request that retrieves the data from IMS. Connection to DB2 is established on the first request for DB2 data.

You can start multiple Gateways concurrently using the SERVERID parameter. The number of instances of the Gateway that you can start concurrently can be from 1 to a value less than or equal to the value set in the **Maximum Connection Count** field in your Network Configuration.

If using the DL/I and BMP interface, you can start only one Gateway task per address space. You use the DRA interface to start multiple Gateways in a single address space. If using the CICS interface, you can start multiple Gateway tasks.

### Starting the Gateway

You can start an instance of the Gateway using one of the following interfaces to IMS:

- DL/I
- BMP
- DRA
- CICS (DB/DC, DBCTL, Local DL/I, or Remote DL/I)–IMS only

### TIBCO Object Service Broker Execution Environments

Service Gateway for IMS/DB executes within the following TIBCO Object Service Broker Execution Environments:

- Batch (DL/I, BMP and DRA)
- Native (DL/I, BMP and DRA)
- CICS–IMS only

Required Startup JCLs

The following table shows the JCL required to start an instance of the Gateway:

| Interface       | In a Batch Execution Environment Use... | In a Native Execution Environment Use... | In a CICS Execution Environment Use... |
|-----------------|---|--|--|
| IMS DL/I        | IMSBDR                                  | IMSEDS                                   | n/a                                    |
| IMS BMP         | IMSBBR                                  | IMSEBS                                   | n/a                                    |
| IMS CICS        | n/a                                     | n/a                                      | IMSCDS                                 |
| IMS DRA         | n/a                                     | IMSEDRA                                  | n/a                                    |
| IMS/DB2<br>DL/I | IM2BDR                                  | IM2EDS                                   | n/a                                    |
| IMS/DB2<br>BMP  | IM2BBR                                  | IM2EBS                                   | n/a                                    |

See Also *TIBCO Object Service Broker Messages With Identifiers* for information on the messages produced by the Gateway.

*TIBCO Object Service Broker for z/OS Installing and Operating* for more information on starting multiple Gateways.

Starting in a Batch or Native Execution Environment Using the DL/I or BMP Interface

If using the DL/I interface, the sample JCL for running in a Batch Execution Environment is in member IMSBDR in the JCL data set; for running in a Native Execution Environment it is in member IMSEDS.

If using the BMP interface, the sample JCL for running in a Batch Execution Environment is in member IMSBBR in the JCL data set; for running in a Native Execution Environment it is in member IMSEBS.

Use IMSEBS (for BMP) and IMSEDS (for DL/I) if you want to start an instance of the Gateway (where the Gateway stays up and users can run their rules from a workbench, edit or browse the IMS table, and so on).

Use IMSBBR (for BMP) and IMSBDR (for DL/I) if you want to start an instance of the Gateway in a batch Execution Environment. In this case, you must supply the rule to execute the user ID, password and library where the rule is located. The Gateway terminates after successful execution of the rule.



In the BMP interface, the sample JCL for running in a Batch Execution Environment is in member IMSBBR in the JCL data set; for running in a Native Execution Environment, in member IMSEBS.

### Considerations When Using the DL/I or BMP Interface

- Do not specify USERTYPE=ONLINE for the Gateway running in a Batch Execution Environment.
- The Execution Environment uses the value of the SERVERPARM parameter specified in the HRNIN DDname as the parm string for DFSRRRC00.
- If you are using the BMP interface, you must define the IMS APPLCTN macro to your IMS online system as described in [Defining the Gateway to the IMS Online System on page 28](#).

## Starting in a TIBCO Object Service Broker CICS Execution Environment Using the CICS Interface

Sample JCL for running the Gateway in a CICS Execution Environment using the CICS interface is in member IMSCDS in the JCL data set. The Gateway tasks are initialized when the CICS Execution Environment starts.

### Considerations When Using the CICS Interface

- You must specify the CICSHURONTRAN=aaaa parameter, where aaaa represents the TIBCO Object Service Broker CICS transaction identifier (trans ID) that the CICS Execution Environment assigns to all CICS transactions associated with the Gateway. Refer to [Supplying Gateway Startup Parameters on page 57](#) for more information. This trans ID must be defined in the CICS Program Control Table and the CICS Process Program Table.
- Each task requires approximately 644 KB plus the POOLSIZE of memory above the line. For example, if the POOLSIZE gateway startup parameter (defined in the IMSSRV00 SYSIN file) is set to the default size of 256 KB, each Gateway task requires 900 KB above the line.
- You can start multiple Gateways concurrently that reside in different CICS Regions. Use the SERVERID gateway startup parameter (defined in the IMSSRV00 SYSIN file) to group together Gateways with common characteristics. The number of instances of the Gateway that you can start concurrently can be from 1 to a value less than or equal to the value set in the **Maximum Connection Count** field in your Network Configuration.
- If you want to access the Gateway from a Batch Execution Environment, you *must* specify the gateway startup parameter USERTYPE=ANY or USERTYPE=LOCAL in the IMSSRV00 SYSIN file.

## Starting in a Batch or Native Environment Using the DRA Interface

If using the DRA interface, the sample JCL for running in a Native Execution Environment is in member IMSEDRA.

### Enabling the DRA Interface

Enabling the DRA interface requires two steps:

1. Copy the DRA Startup/Router routine (DSFPRRC0) into the Gateway's load library.

The Gateway must load DSFPRRC0. The system programmer can copy the routine from IMS.RESLIB library or can concatenate the IMS.RESLIB library to the TIBCO Object Service Broker Gateway step library.

2. The system programmer must place the DFSPZPxx load module (DRA startup table) in a Gateway load library.

On the initialization request, the Gateway specifies the startup table name suffix (DRASUFFIX=xx). The default load module, DFSPZP00, contains default values for the DRA initialization parameters. If values other than the defaults are required, the system programmer can modify the module (naming it DFSPZPxx), assemble and link it, and load it into the TIBCO Object Service Broker load library. See the current *IMS Customization Guide* IBM manual for more information.

## Starting to Access Both IMS and DB2 Data in a TIBCO Object Service Broker Batch or Native Execution Environment Using the DL/I Interface

If using the DL/I interface, the sample JCL for running in a Batch Execution Environment is in member IM2BDR in the JCL data set; for running in a Native Execution Environment it is in member IM2EDS.

### Considerations When Using the DL/I Interface

- Do not specify USERTYPE=ONLINE for an instance of the Gateway concurrently accessing DB2 data running in a Batch Execution Environment.
- The Execution Environment uses the value of the SERVERPARM parameter specified in the HRNIN DDname as the parm string for DFSRRRC00.
- The DDITV02 DD statement specifies the data set that contains the input parameters used by the IMS subsystem attach facility for DB2. Member XIM2DB21 in the CNTL data set contains sample input parameters:

```
$DB2SSI$,SYS1,DSNMIN10,,R,-,$JOBNAME$, $PLNAME$,S6BDRSSR
```

where

|                    |   |
|--------------------|---|
| <i>\$DB2SSI\$</i>  | The DB2 subsystem identifier. This OSEMOD parameter must be modified.   |
| <i>SYS1</i>        | The language interface token value.   |
| <i>DSNMIN10</i>    | The DB2 initialization module name.   |
| <i>null</i>        | The resource translation table. If no RTT is required, this parameter must be set to null.  |
| <i>R</i>           | The region error option. When set to R, it returns any SQL return codes to the Service Gateway for IMS program.   |
| <i>-</i>           | The DB2 command recognition character.  |
| <i>\$JOBNAME\$</i> | The connection name that represents the name of the job step that is the coordinator for the DB2 activity. This OSEMOD parameter must be modified.<br><br>If not specified, the connection name defaults to the job name (if the Gateway is running as a batch job) or the started task job name (if the Gateway is running as a started task). |
| <i>\$PLANAME\$</i> | The Gateway plan name. This OSEMOD parameter must be modified.  |
| <i>S6BDRSSR</i>    | The Gateway program name. Do not modify this parameter.   |

- If the gateway parameter FSLEVEL=1, the PLAN defined in the DDITV02 statement must be rebound with the current TIBCO Object Service Broker DBRMLIB library.
- The DDOTV02 DD statement specifies an output data set used to receive messages from the IMS subsystem attach facility about in-doubt and diagnostic information.

**See Also**     The *DB2 Application Programming & SQL Guide* IBM manual for information about DB2 DL/I batch support. See *TIBCO Object Service Broker for z/OS Installing and Operating* for more information on starting multiple Gateways concurrently.

## Starting to Access Both IMS and DB2 Data in a TIBCO Object Service Broker Batch or Native Execution Environment Using the BMP Interface

If using the BMP interface, the sample JCL for running in a Batch Execution Environment is in member IM2BBR in the JCL data set; for running in a Native Execution Environment it is in member IM2EBS.

### Considerations When Using the BMP Interface

- Do not specify USERTYPE=ONLINE for an instance of the Gateway running in a Batch Execution Environment.
- The Execution Environment uses the value of the SERVERPARM parameter specified in the HRNIN DDNAME as the parameter string for DFSRRC00.
- If you are using the BMP interface, you must define the IMS APPLCTN macro to your IMS online system as described in [Defining the Gateway to the IMS Online System on page 28](#).
- Include the SSM parameter in the startup JCL, in the IMS Control Region startup JCL, or in both.
- Specify the DB2 subsystem to which the IMS Control Region is to connect in the IMS PROCLIB Subsystem Specification Member (SSM).
- If you do not include the PLAN parameter, the DB2 PLAN name defaults to the PSB name.

See Also     The *DB2 Application Programming & SQL Guide* IBM manual for information about DB2 support.

# Dynamically Modifying Gateway Tasks

## Dynamically Starting Gateway Tasks

If the number of Gateway tasks is insufficient to process transaction requests, unsatisfied requests are queued until a free Gateway is available. You can dynamically increase the number of Gateway tasks without restarting the Execution Environment using the **MODIFY** operator command.

- From the z/OS operator console, the command format is:  
**MODIFY ee\_jobname , STARTNUMSERVER=nn , TYPE=IMS or IM2**
- From the z/OS CICS operator console, the command format is:  
**MODIFY ee\_jobname ,HOPS STARTNUMSERVER=nn , TYPE=IMSCICS**
- For CICS authorized users, the command format is:  
**HOPS STARTNUMSERVER=nn , TYPE=IMSCICS**

where

|                   |  |
|-------------------|--|
| ee_jobname        | The name of the batch job under which the Execution Environment is running.  |
| nn                | The number of Gateway tasks to start. This number can be from 1 to a value less than or equal to the value set in the <b>Maximum Connection Count</b> field in your Network Configuration. |
| IMS, IM2, IMSCICS | IMS=IMS database access only.<br>IM2=Concurrent access to IMS and DB2 data.<br>IMSCICS=Running under CICS DL/1.  |

## Dynamically Setting Gateway Tasks

You can also use the **MODIFY** operator command to dynamically set the maximum number of Gateway tasks available in a particular Execution Environment.

- From the z/OS operator console, the command format is:  
**MODIFY ee\_jobname , SETNUMSERVER=nn , TYPE=IMS or IM2**
- From the z/OS CICS operator console, the command format is:

**MODIFY** *ee\_jobname* ,**HOPS SETNUMSERVER=nn ,TYPE=IMSCICS**

- For CICS authorized users, the command format is:

**HOPS SETNUMSERVER=nn ,TYPE=IMSCICS**

where

|                          |   |
|--------------------------|---|
| <i>ee_jobname</i>        | The name of the batch job under which the Execution Environment is running.   |
| <i>nn</i>                | The maximum number of Gateway tasks available for this particular Execution Environment. This number can be from 1 to a value less than or equal to the value set in the <b>Maximum Connection Count</b> field in your Network Configuration. |
| <i>IMS, IM2, IMSCICS</i> | IMS=IMS database access only.<br>IM2=Concurrent access to IMS and DB2 data.<br>IMSCICS=Running under CICS DL/1.   |

See Also *TIBCO Object Service Broker for z/OS Installing and Operating* for more information on the **MODIFY** operator command and TIBCO Object Service Broker Network Configuration.

# Shutting Down the Gateway

You can shut down a Gateway using the **MODIFY** operator command from the z/OS operator console, or by using the RESOURCE MANAGEMENT option from the Administration menu.



These methods only shut down the Gateway. To shut down the Execution Environment, which automatically shuts down all Gateways running in that Execution Environment, refer to *TIBCO Object Service Broker for z/OS Installing and Operating*.

## MODIFY Operator Command for Data Object Broker

The format of the **MODIFY** command is:

**MODIFY** *dob\_jobname* , **STOPSERVER**=*idprefix*

|                    |   |
|--------------------|---|
| <i>dob_jobname</i> | Name of the batch job under which the Data Object Broker is running.  |
| <i>idprefix</i>    | Unique name of the TIBCO Service Gateway for IMS/DB. The IDPREFIX parameter is specified in the Gateway startup JCL. The default is IMS01. To view the unique name assigned to existing Gateways, select the SERVICE PROVIDER ACTIVITY option from the Administration menu. |

You can also use one of three variations of the **MODIFY** operator command to shut down groups of instances of the Gateway.

- Shut down all Gateways:

**MODIFY** *dob\_jobname* , **STOPSERVER**=**ALLIMS** or **ALLIM2**

ALLIMS is for gateways accessing only IMS data, ALLIM2 is for gateways accessing both IMS and DB2 data.

- Shut down all Gateways with a common IDPREFIX:

**MODIFY** *dob\_jobname* , **STOPSERVER**=*idprefix*\*

- Shut down all Gateways with a common SERVERID:

**MODIFY** *dob\_jobname* , **STOPSERVER**=**SRVID***serverid*

## MODIFY Operator Command for the Execution Environment

You can use the **MODIFY** operator command to shut down any Gateway running in the Execution Environment (or under CICS DL/I).

- From the z/OS operator console, the command format is:  
**MODIFY ee\_jobname ,STOPNUMSERVER=1 ,TYPE=IMS or IM2**
- From the z/OS CICS operator console, the command format is:  
**MODIFY ee\_jobname ,HOPS STOPNUMSERVER=1 ,TYPE=IMSCICS**
- For CICS authorized users, the command format is:  
**HOPS STOPNUMSERVER=1 ,TYPE=IMSCICS**

where

|                          |   |
|--------------------------|---|
| <i>ee_jobname</i>        | Name of the batch job under which the Execution Environment is running.   |
| <i>nn</i>                | The number of new instances of the Gateway to stop. Since only one Gateway runs in an Execution Environment at a time, you can stop only one Gateway. |
| <i>IMS, IM2, IMSCICS</i> | IMS=IMS database access only.<br>IM2=Concurrent access to IMS and DB2 data.<br>IMSCICS=Running under CICS DL/1.                                       |

## RESOURCE MANAGEMENT Option

Use the **RESOURCE MANAGEMENT** option from the Administration menu to shut down either one Gateway or groups of Gateways. For details on the Administration menu, see *TIBCO Object Service Broker for z/OS Installing and Operating*.



## Chapter 2      **Operating the Service Gateway for IMS/DB**

This chapter describes how to operate Service Gateway for IMS/DB.

### Topics

---

- [Defining IMS Databases to TIBCO Object Service Broker, page 50](#)
- [Binding IMS Table Definitions, page 55](#)
- [Preparing the IMS and DB2 Environments for Accessing DB2 Data, page 56](#)
- [Supplying Gateway Startup Parameters, page 57](#)
- [Dynamically Changing Gateway Parameters, page 74](#)
- [Implementing External Security, page 76](#)
- [Implementing Fail Safe Processing, page 84](#)
- [Understanding Other Operational Procedures, page 86](#)
- [Connecting the Gateway to a Windows or Solaris Data Object Broker, page 92](#)

## Defining IMS Databases to TIBCO Object Service Broker

---

To define IMS databases to TIBCO Object Service Broker, you must:

1. [Extract IMS database information, page 50](#)
2. [Implement a security strategy, page 51](#)
3. [Load an IMS database definition into TIBCO Object Service Broker, page 51](#)

These tasks are described in the sections below.

### Task A Extract IMS database information

The S6BIMSU program extracts IMS database, segment, and field information for physical databases from the IMS DBD load modules. S6BIMSU can extract only database and partial segment information for logical databases. Refer to [Completing the Definition for a Logical Database on page 53](#) for more information. A SYSIN member identifies the DDname of the IMS DBD load libraries and specifies the load modules to be processed by S6BIMSU. The extracted information is stored in the data set associated with the SYSPUNCH DDname.

You must allocate the SYSPUNCH data set before you run S6BIMSU. You can use an existing partitioned data set and specify a new member. If you specify an existing member name, its contents are overwritten.

[Task C, Load an IMS database definition into TIBCO Object Service Broker, on page 51](#) describes how to define the SYSPUNCH data set to a TIBCO Object Service Broker import file. The IMS tool loads DBD information into control tables defining the IMS database to TIBCO Object Service Broker.

### Sample JCL for Running the S6BIMSU Program

Sample JCL for the S6BIMSU program is included as member IMSUJCL in the JCL data set. Modify this JCL as required.

### Sample SYSIN Member for S6BIMSU

A sample SYSIN member is located in member XIMSUJCL of the CNTL data set. This SYSIN member identifies the DBD load modules to be processed and the DDnames associated with the load libraries.

The DBD statement must start in column 5. One or more load libraries and modules can be specified in the SYSIN member. Use the following syntax when defining the SYSIN member:

**DBD**(*dbdname*, *library*)

where

*dbdname* – the DBD load module name.

*library* – the DDname of the DBD load library.

### Task B Implement a security strategy

Before loading the extracted data set into TIBCO Object Service Broker, users responsible for loading the DBD information must be given MODIFY\_DEFN authority to the IMSGEN import table. Only a level-7 system administrator can assign MODIFY\_DEFN for the import table.

See Also *TIBCO Object Service Broker Managing Security* for more information on security levels.

### Task C Load an IMS database definition into TIBCO Object Service Broker

To load the information extracted from the IMS DBD using the S6BIMSU program you must:

1. Identify the data set that contains the extracted IMS DBD information to TIBCO Object Service Broker.
2. Use the IMS tool to load the IMS DBD information.

These steps are described in this task.

### Importing IMS Database Definitions

The S6BIMSU program creates a SYSPUNCH data set. Information from this data set must be imported into TIBCO Object Service Broker where it is stored in the @IMSDBS, @IMSSEGS, @IMSDBD, and @IMSSIXFIELDS control tables. To import the SYSPUNCH information, complete the following steps:

1. Using the Table Editor, edit the IMSGEN occurrence of the IMPORTFILES table.

In the **FILENAME** field of this table, specify the name of the output data set created using the S6BIMSU program. If the data set is partitioned, include the member name.

2. Execute IMS from the EX execute rule option or the primary command field on the workbench.

The screen below shows the Manager Utilities for IMS Data menu. From this screen you can select an option by typing an **S** beside the option and pressing Enter.

---

Command ===>

MANAGER UTILITIES FOR IMS DATA

-----

Enter an "S" to select a function

- \_ Import IMS Database Definitions
- \_ List IMS Databases
- \_ IMS DBD Extract Utility

PFKEYS: 1=HELP 3=EXIT 12=EXIT

---

3. Select the Import IMS Database Definitions option.

A message similar to the following appears at the bottom of the screen:

IMSGEN filename will be imported; PRESS ENTER or type 'confirm'

4. Press Enter to confirm that you want to import the definitions.

If any definitions are currently defined to TIBCO Object Service Broker, they are overwritten. The existing IMS table definitions using these databases still exist; however, they do not contain new DBD information.

5. Using the Table Definer, modify the IMS table definitions that require the new DBD information.

6. If the target segment of an IMS secondary index is not the root segment, and you use this secondary index to access the IMS database, IMS returns the segments in an order different from the expected order

If this type of secondary index is used to access IMS data from TIBCO Object Service Broker, you must do one of the following:

- Modify the **PARENT** and **LEVEL** fields of the @IMSSEGS control table to reflect the new order in which the segments are returned.
- Create an IMS logical database that reflects the new order in which the segments are returned and define the logical database to TIBCO Object Service Broker.



If you require access to an IMS database from TIBCO Object Service Broker using both the primary and secondary processing sequence, a logical database is required. Refer to [Defining Gateway Program Specification Blocks on page 26](#) for more information.

## Completing the Definition for a Logical Database

Logical databases establish relationships between segments that belong to other physical databases. Logical database definitions refer only to the physical databases for each segment named; therefore, the S6BIMSU program extracts only the following logical DBD information:

- Database name (for @IMSDBS)
- Database type (for @IMSDBS)
- Segment names (for @IMSSEGS)
- Segment level within the hierarchy (for @IMSSEGS)
- Name of segment's parent (for @IMSSEGS)

You must add the missing information to the @IMSSEGS, @IMSDBD, and @IMSSIXFIELDS control tables before you can access the logical database from TIBCO Object Service Broker. You can add the information manually or complete the following steps:

1. Using the S6BIMSU program, import the physical database definition for each segment named in the logical database definition.
2. Check that the physical database segment names are the same as the logical database segment names.

If the names are not the same, note which physical database segment name corresponds to each logical database segment name.

3. Using the CT copy table option from the workbench, copy the @IMSSEGS(*physical database name*) table instance to the @IMSSEGS(*logical database name*) table instance.

This provides information for the following fields:

- LENGTH
- KEYNAME
- VARSEG
- MINLENGTH
- UNIQUESEQUENCE

4. If you have a corresponding logical database, do one of the following, depending on whether your database definition uses secondary indexes:
  - If your database definition does not use secondary indexes, use the CT copy table option from the workbench to copy the @IMSDBD(*physical database name, physical segment name*) table instance to the @IMSDBD(*logical database name, logical segment name*) table instance. This defines the field information. The *physical segment name* is the name that corresponds to the *logical segment name*, described in Step 2.



If the logical database is made up of segments from different databases, check the logical database definition to determine the order in which the segments are defined, and copy them into the @IMSDBD(*logical database name, logical segment name*) table instance in this same order. Ensure that the **IMSOFF** field is updated to reflect the correct offsets after combining the physical segment definitions.

- If your database definition does use secondary indexes, use the CT copy table option from the workbench to copy the @IMSSIXFIELDS(*physical database name, physical targetsegment name, physical field name*) table instance to the @IMSSIXFIELDS(*logical database name, logical targetsegment name, logical field name*) table instance.



Existing control tables are overwritten by S6BIMSU. After using this program, you must edit the @IMSSEGS and @IMSDBD tables as described above to replace the additional information required for logical databases.

## Binding IMS Table Definitions

---

You can bind an IMS table definition but not its data. IMS tables for which you request binding are bound to both the Execution Environment and the Gateway when they are accessed from a rule.

### Rebinding

If you change a definition, it is automatically rebound in the Gateway.

### Specifying Available Gateway Space Using the POOLSIZE Parameter

You can specify the maximum amount of space available to hold all IMS table definitions by using the POOLSIZE gateway startup parameter. Refer to [Supplying Gateway Startup Parameters on page 57](#) for more information.

**See Also**     *TIBCO Object Service Broker Application Administration* for information on binding tables.

## Preparing the IMS and DB2 Environments for Accessing DB2 Data

---

Before you can access DB2 data, ensure you properly modified the IMS and DB2 environments as described in [Preparing for Installation on page 6](#) and *TIBCO Service Gateway for DB2 Installing and Operating*. In addition, if a Static SQL handler is used by the Gateway, add the default AMODE and RMODE to the link control card input statements: AMODE(31) and RMODE(24)

### Changes to Link Control Card Members

This example illustrates the changes you must make to each link control card member generated by @STATICSQL

---

```
INCLUDE OBJECT2(D2CSGEMP)
MODE AMODE(31),RMODE(24)
ENTRY D2CSGEMP
NAME      D2CSGEMP(R)
```

---



# Supplying Gateway Startup Parameters

You can specify gateway parameters in the HRNIN DD statement, in the startup JCL, or in the IMSSRV00 SYSIN file. For more information, refer to [Specifying Gateway Parameters on page 34](#).



The PSBNAME and SERVERID parameters can be overridden at runtime. Refer to [Dynamically Changing Gateway Parameters on page 74](#) for more information.

## HRNIN Parameters

The gateway parameters you specify in the HRNIN DD statement are:

|               |   |
|---------------|---|
| CICSHURONTRAN | Valid only for an instance of the Gateway using the CICS Interface. This is the TIBCO Object Service Broker transaction identifier (trans ID) that the CICS Execution Environment assigns to all CICS transactions associated with the Gateway. It is recommended that you specify HURN.  |
| DRASUFFIX     | Specifies the suffix for the DRA startup parameter module. Required if IMSDRA is specified. Refer to <a href="#">Starting in a Batch or Native Environment Using the DRA Interface on page 42</a> for more information.   |
| MDL           | The pattern for selecting the VTAM ACB name that the Gateway uses for communications. If not specified, the Gateway uses the TDS parameter as the pattern. If you do not specify this parameter, ensure that the TDS parameter is a valid VTAM ACB model. For example, if TDS=OSB001 and you do not include this parameter, the model OSBnnn is used. |

---

|                |   |
|----------------|---|
| <b>SERVERS</b> | The number of Gateway tasks to initialize. For Gateways using the DL/I or BMP interface the value must be 1 or not supplied. For Gateways using the DRA and CICS interface the default is 1. This number can be from 1 to a value less than or equal to the value set in the <b>Maximum Connection Count</b> field in your Network Configuration. For more information, refer to the <i>TIBCO Object Service Broker for z/OS Installing and Operating</i> manual. |
|----------------|---|

---



You can dynamically increase the number of Gateway tasks using the **MODIFY** operator command. Refer to [Dynamically Modifying Gateway Tasks on page 45](#) for more information.

---

|                   |   |
|-------------------|---|
| <b>SERVERPARM</b> | Valid only for an instance of the Gateway using the DL/I or BMP interface. The parameter string must be passed to DFSRRC00. The default for an instance of the Gateway accessing only IMS data is 'DLI,S6BDRSSR,S6BIMSS,,,,,,,,,N,,,Y'. The default for an instance of the Gateway concurrently accessing IMS and DB2 data is 'DLI,DSNMTV01,S6BIMSS,,,,,,,,,N,,,Y'. |
|-------------------|---|

---

---

|                   |   |
|-------------------|---|
| <b>SERVERTYPE</b> | Specifies the type of Gateway to run. The default is IMS. Valid specifications: |
|-------------------|---|

---

|         |  |
|---------|--|
| IMS     | Gateway running in a Batch Execution Environment and using the DL/I or BMP interface to IMS.   |
| IM2     | Gateway for IMS concurrently accessing IMS and DB2 data running in a Batch Execution Environment and using the DL/I or BMP interface to IMS. |
| IMSDFS  | Gateway running in a Native Execution Environment and using the DL/I or BMP interface to IMS.  |
| IM2DFS  | Gateway running in a Native Execution Environment, concurrently accessing IMS and DB2 data, and using the DL/I or BMP interface to IMS.      |
| IMSCICS | Gateway running in a CICS Execution Environment and using the CICS interface to IMS.   |
| IMSDRA  | Gateway running in a native or batch Execution Environment using the DRA interface.  |

|            |   |
|------------|---|
| <b>TDS</b> | Supplies the communications identifier of the Data Object Broker with which the Gateway communicates. This parameter is required and does not have a default. |
|------------|---|

## IMSSRV00 Parameters

The Gateway parameters you specify in IMSSRV00 are:

|              |   |
|--------------|---|
| <b>CLASS</b> | Has meaning only if SECLEVEL=1. Specifies the System Authorization Facility (SAF) class name to which the RESOURCE parameter value belongs. Valid entries are from 1 to 8 bytes long. This name and the name specified for the RESOURCE parameter are used by the Gateway when issuing the SAF RACROUTE REQUEST=AUTH macro to determine if the TIBCO Object Service Broker session ID, current security group, or a combination of both has access to the resource-level security. Refer to <a href="#">Implementing External Security on page 76</a> for more information. |
|--------------|---|

|                |   |     |                              |    |  |
|----------------|---|-----|------------------------------|----|--|
| DBDCLASS       | Has meaning only if SECLEVEL=1. The SAF security class under which the IMS databases are secured and defined.   |     |                              |    |  |
| DEBUG          | Causes debugging messages to be echoed on the Gateway and placed in the system message log.   |     |                              |    |  |
| DECOUPL        | Has meaning only if SECLEVEL=1. The default is NO. This parameter is documented in the <i>RACF External Security Interface (RACROUTE) Macro Reference</i> IBM manual for z/OS and VM. Depending on the external security package you are using, specify one of the following: <table><tr><td>YES</td><td>RACF security is being used.</td></tr><tr><td>NO</td><td>Security such as CA-ACF2 or CA-Top Secret is being used. REQSTOR and SUBSYS are used to verify SAF control point and subsystem. Review the defaults for these parameters and modify if required.</td></tr></table>  | YES | RACF security is being used. | NO | Security such as CA-ACF2 or CA-Top Secret is being used. REQSTOR and SUBSYS are used to verify SAF control point and subsystem. Review the defaults for these parameters and modify if required. |
| YES            | RACF security is being used.  |     |                              |    |  |
| NO             | Security such as CA-ACF2 or CA-Top Secret is being used. REQSTOR and SUBSYS are used to verify SAF control point and subsystem. Review the defaults for these parameters and modify if required.  |     |                              |    |  |
| DLIUPDATELIMIT | <p>Use this parameter only in development or if you enable ad hoc access to IMS through TIBCO Object Service Broker to avoid IMS database scans that overload IMS lock pools.</p> <p>Controls the number of update DL/I calls the Gateway can perform within a DL/I unit of work. If the limit is exceeded, the Gateway returns a COMMITLIMIT error with the appropriate message. After a COMMITLIMIT error the only valid request is a COMMIT/ROLLBACK; any other request terminates the DL/I unit of work with a ROLLBACK and the Gateway returns a SERVERFAIL. IMS calls GHU, GHN, and ISRT are counted in the update limit. The default is 0. Valid values:</p> |     |                              |    |  |

|               |   |
|---------------|---|
| 0             | Compatible with current Gateway functionality.  |
| 1 - 2**31 - 1 | Limit on number of GHU, GHN, and ISRT calls within one TIBCO Object Service Broker transaction. |

**DLIQUERYLIMIT**

Use this parameter only in development or if you enable ad hoc access to IMS through TIBCO Object Service Broker to avoid IMS database scans that overload IMS lock pools.

Controls the number of query DL/I calls the Gateway can perform within a DL/I unit of work. If the limit is exceeded, the Gateway returns a COMMITLIMIT error with the appropriate message. After a COMMITLIMIT error the only valid request is a COMMIT/ROLLBACK; any other request terminates the DL/I unit of work with a ROLLBACK and the Gateway returns a SERVERFAIL. IMS calls GU and GN are counted in the query limit. A GU or GN call acquires an IMS record lock if PCB procopt = a/i/r/d (update). The lock is released after the application removes positioning from the record unless the record is updated. The default is 0. Valid values:

|               |  |
|---------------|--|
| 0             | Compatible with current Gateway functionality. |
| 1 - 2**31 - 1 | Limit on number of GHU, GHN, and ISRT calls.   |

**EXTERNALGROUP**

Has meaning only if SECLEVEL=1 and EXTERNALUSERID=USERID. Specifies the use of both the TIBCO Object Service Broker session ID and current security group name to verify access to IMS resources. Do not include this parameter if EXTERNALUSERID=GROUP. If the value of the current security group exceeds 8 bytes, only the TIBCO Object Service Broker session ID is used. Refer to [Establishing the TIBCO Object Service Broker Authorization ID on page 82](#) for more information.

|                |  |
|----------------|--|
| EXTERNALUSERID | Has meaning only if SECLEVEL=1. Specifies the use of the TIBCO Object Service Broker session ID or the current security group name to verify access to IMS resources. The default is USERID. Refer to <a href="#">Establishing the TIBCO Object Service Broker Authorization ID on page 82</a> for more information. |
| USERID         | The TIBCO Object Service Broker session ID.  |
| GROUP          | The current security group. This group can be up to 16 characters, but only 8 characters are supported. A SECURITYFAIL is signalled for more than 8 characters.  |
| FSLEVEL (FSL)  | Use to specify the level of Fail Safe processing. The default value is zero.<br><br>Valid values:  |

- 1 Activate Fail Safe Level 1. The Gateway informs the Data Object Broker that it can support Fail Safe level-1 processing. If the Gateway is to attach to a z/OS Data Object Broker, the Data Object Broker's connection attribute setting "commit level" must be set to 1. If not, the Gateway connection is rejected. Refer to [Implementing Fail Safe Processing on page 84](#) for more information.

You must specify the TRXDB, FSTABLENAME, RECOVERYID, and RECOVERYPASSWORD parameters.

- 0 De-activate Fail Safe processing. The Gateway informs the Data Object Broker that it does not support Fail Safe level-1 processing. If the Gateway is to attach to a z/OS Data Object Broker, the Data Object Broker's connection attribute setting "commit level" must be set to 0. If not, the Gateway connection is rejected. Refer to [Implementing Fail Safe Processing on page 84](#) for more information.

**FSTABLENAME**

Required only if FSLEVEL=1. Specifies the name of the DB2 table that maps to the DB2 or IMS transaction database. The default is @IMSFSTRXDB for IMS and @DB2FSTRXDB for concurrent access to IMS and DB2 data.



If you also specify the TRXDB parameter, the system uses the TRXDB *userid.trxdb* and replaces the DB2 *userid* and *trxdb* in the DB2 table specified in FSTABLENAME.

If you have multiple instances of the Gateway with the same SERVERID, the FSLEVEL, FSTABLENAME, and TRXDB gateway parameters must have the same values for each instance.

**IDPREFIX**

Specifies a unique name for each instance of the Gateway. Each Gateway must have a unique IDPREFIX and it must be a valid level-1 TIBCO Object Service Broker user ID. The Gateway uses this parameter as the Gateway connection identifier when connecting to the Data Object Broker. This parameter can have up to eight characters. The default is IMS01.



If you have multiple instances of the Gateway using the CICS interface and connecting to the same Data Object Broker, ensure the first five characters of the IDPREFIX are unique.

If USERTYPE=LOCAL, the Gateway appends four decimal digits to the first four characters of the prefix.

|          |  |
|----------|--|
| MODE     | Determines whether the Gateway operates in update (UPD) or read-only (RO) mode. If MODE=RO, update requests are rejected. The default is UPD.  |
| PLAN     | Provides the DB2 plan name if using the BMP interface. If omitted, the DB2 PLAN name defaults to the PSBNAME. For more information, refer to <i>TIBCO Service Gateway for DB2 Installing and Operating</i> .   |
| POOLSIZE | Sets the amount of space (in KB) to hold IMS or DB2 table definitions in the Gateway, to a maximum of 16384 KB. The default value is 256. An estimate of the number of IMS or DB2 tables that can be accessed in a single transaction is POOLSIZE divided by CTABLESIZE. Refer to <a href="#">Estimating the CTABLESIZE Parameter on page 72</a> for more information.             |
| PSBCLASS | Has meaning only if SECLEVEL=1. The SAF security class under which the IMS Program Specification Blocks (PSBs) are secured and defined.  |
| PSBNAME  | <p>Provides the Gateway with the default PSB name. This PSB is scheduled if no other value can be ascertained from the transaction. The default is S6BIMSS.</p> <p>Refer to the THREADUSAGE parameter (below) for more information.</p> <p>The PSBNAME parameter can be overridden at runtime. Refer to <a href="#">Table Type Attributes on page 74</a> for more information.</p> |



**RECOVERYID**

Has meaning only if FSLEVEL=1 *and* SECLEVEL=1. Sets the TIBCO Object Service Broker authorization ID that you want to use during recovery when querying the IMS transaction database to see if the transaction completed (in-doubt transactions). This parameter can be up to eight characters. The default is the ID used to start the Gateway.



If you have multiple instances of the Gateway with the same SERVERID, ensure that the FSLEVEL, FSTABLENAME, RECOVERYID, and TRXDB gateway parameters have the same values for each Gateway.

**REQSTOR**

Has meaning only if SECLEVEL=1 *and* DECOUPL=NO. The default is S6BIMSS. Specifies the SAF control point name to use when issuing SAF security calls. The value specified identifies the program or piece of code issuing the request.

The REQSTOR parameter is documented in the *External Security Interface (RACROUTE) Macro Reference* IBM manual for z/OS and VM.

**RESOURCE**

Has meaning only if SECLEVEL=1. Specifies the name of the resource such as the IMS transaction, a group of transactions, the PSB name, an Application Group Name (AGN), or any other site specific value that uniquely identifies the Gateway session. Valid entries are from 1 to 8 bytes long.

Using the names specified for this and the CLASS parameter, the Gateway issues the SAF RACROUTE REQUEST=AUTH macro. This determines if the TIBCO Object Service Broker session ID, the current security group, or a combination of the two has access to the specified class/resource combination. Refer to [Implementing External Security on page 76](#) for more information.

**RESPONSEMODE**

The mode that the Gateway uses to respond to requests. You can use this mode to free up a Gateway more quickly for a new transaction in read-only situations. For use only with a Data Object Broker on z/OS. The default is ASYNC. Valid entries:

|       |   |
|-------|---|
| ASYNC | The Data Object Broker releases the Gateway from the TIBCO Object Service Broker transaction when the end of transaction request is sent, provided no updates are requested. The next transaction can then be scheduled before the previous transaction is completed. ASYNC is recommended for online Gateways. If SYNC is used, performance can be affected. |
| SYNC  | Causes the Data Object Broker to wait for the Gateway to complete end of transaction processing. SYNC is recommended for Gateways running in batch mode.  |

|       |   |
|-------|---|
| SCOPE | The length of time to hold the connection between an instance of the Gateway and a IMS subsystem. The defaults are SESSION for a Gateway running in a non-CICS environment and TRANSACTION for a Gateway running in a CICS environment. |
|-------|---|

|             |  |
|-------------|--|
| SESSION     | Hold the connection for the life of the Gateway. The connection is not established until the first request for IMS data is processed by the Gateway. |
| TRANSACTION | Hold the connection for the life of a TIBCO Object Service Broker transaction.   |
| <i>nnnn</i> | Hold the connection for <i>nnnn</i> seconds beyond the end of the TIBCO Object Service Broker transaction. Valid entries are 0-9999.                 |

|          |   |
|----------|---|
| SECLEVEL | Specifies the level of authorization to use when accessing IMS data. The default value is 0. Valid entries: |
|----------|---|

|   |   |
|---|---|
| 0 | Refer to the table under <a href="#">Implementing IMS Security on page 31</a> .   |
| 1 | <p>The TIBCO Object Service Broker session ID, current security group, or a combination of the two is used to verify access to the specified combinations of CLASS, RESOURCE, PSBCLASS, DBDCLASS, SEGCLASS, IMS PSBs, IMS databases, and IMS databases/segments. Refer to <a href="#">Implementing External Security on page 76</a> for more information.</p> <p>If SECLEVEL=1 and FSLEVEL=1, review the default for the RECOVERYID parameter.</p> <p>If SECLEVEL=1, review the defaults for the EXTERNALUSERID and EXTERNALGROUP parameters. Refer to <a href="#">Establishing the TIBCO Object Service Broker Authorization ID on page 82</a> for more information.</p> |

|                 |  |
|-----------------|--|
| <b>SEGCLASS</b> | Has meaning only if SECLEVEL=1. The SAF security class under which IMS databases/segments, and PSBCLASS and PSBs are secured.  |
| <b>SERVERID</b> | <p>Identifies a pool of Gateways with common characteristics. If you have multiple instances of the Gateway with the same SERVERID, ensure that the FSLEVEL, FSTABLENAME, RECOVERYID, and TRXDB gateway parameters have the same values for each Gateway. This parameter can be up to eight characters long. The default is DEFAULT.</p> <p>This parameter can be overridden at runtime. Refer to <a href="#">Dynamically Changing Gateway Parameters on page 74</a> for more information.</p> |
| <b>SUBSYS</b>   | <p>Has meaning only if SECLEVEL=1 and DECOUPL=NO. Specifies the subsystem name for the SAF security call. The default is HURON.</p> <p>This parameter is documented in the <i>External Security Interface (RACROUTE) Macro Reference</i> IBM manual for z/OS and VM.</p>   |

---

|                    |  |
|--------------------|--|
| <b>THREADUSAGE</b> | Valid only for an instance of the Gateway executing in a CICS Execution Environment. Specifies where to get the PSB name and whether to check the PSB name specified with each IMS table accessed in a TIBCO Object Service Broker transaction. The default is TRANSACTION. Valid entries: |
|--------------------|--|

---

|             |   |
|-------------|---|
| SESSION     | <p>The PSB specified by the PSBNAME parameter in the IMSSRV00 SYSIN file is used.</p> <p>If no PSBNAME is specified in IMSSRV00, S6BIMSS is used.</p> <p>If a PSB is specified in an IMS table definition, it is ignored.</p>   |
| TRANSACTION | <p>The PSB specified by the first IMS table accessed in a TIBCO Object Service Broker transaction is used.</p> <p>If the first IMS table accessed in a transaction does not contain a PSB, the PSB specified by the PSBNAME parameter in the IMSSRV00 SYSIN file is used.</p> <p>If no PSBNAME is specified in IMSSRV00, S6BIMSS is used.</p> <p>When a PSB is established, it is used for all subsequent TIBCO Object Service Broker IMS access requests for the duration of the transaction.</p>      |
| TABLE       | <p>The PSB specified by the first IMS table accessed in a TIBCO Object Service Broker transaction is used.</p> <p>If the first IMS table accessed in a transaction does not contain a PSB, the PSB specified by the PSBNAME parameter in the IMSSRV00 SYSIN file is used.</p> <p>If no PSBNAME is specified in IMSSRV00, S6BIMSS is used.</p> <p>All remaining IMS tables accessed in the transaction must specify either the same PSB or no PSB. If there is a discrepancy, the transaction fails.</p> |

|       |  |
|-------|--|
| TRACE | <p>Logs the Segment Search Arguments (SSAs) built by TIBCO Object Service Gateway for IMS/DB and sent to the CICS Execution Environment. It also includes additional information similar to the DL/I Trace Facility. To activate the TRACE you must configure the following:</p> <ul style="list-style-type: none"> <li>• Allocate a fixed block data set, LRECL=132.</li> <li>• Add a DD statement for the fixed block data set to the CICS Region with DDname HIMTRACE.</li> </ul> |
|-------|--|



Include the TRACE parameter for error resolution only. For Service Gateway for IMS, the TRACE parameter also turns on SQL tracing in the HDBTRACE DD. For more information, refer to *TIBCO Service Gateway for DB2 Installing and Operating*.

---

|       |  |
|-------|--|
| TRXDB | <p>The transaction database. Required only if FSLEVEL=1. For access to IMS data only, this parameter specifies the name of the IMS transaction database.</p> <p>For concurrent access to IMS and DB2 data, it is required only if the actual Fail Safe DB2 transaction table name and creator are different than the ones that the DB2 table name specified using the FSTABLENAME parameter points. The name of the DB2 transaction file to contain information about the last transaction processed by the Gateway.</p> |
|-------|--|

---



If you have multiple instances of Gateway with the same SERVERID, the FSLEVEL, FSTABLENAME, RECOVERYID, and TRXDB gateway parameters must have the same values for each Gateway.

---

|          |  |
|----------|--|
| USERTYPE | <p>The type of user that can connect to this Gateway. When allocating an instance of the Gateway, the Data Object Broker first tries to allocate an online user to an online Gateway. If it cannot, it allocates the user to a Gateway with USERTYPE=ANY. For use only with a Data Object Broker on z/OS. Valid entries:</p> |
|----------|--|

---

|        |   |
|--------|---|
| ANY    | Enables batch and online users to use the Gateway. This is the default for Gateways running in a Batch or Native Execution Environment.   |
| LOCAL  | <p>Causes the SERVERID to be replaced by a unique identifier (the Gateway jobname). Reserves the Gateway for user sessions executing in a Batch or CICS Execution Environment as the Gateway. Not valid for online Gateways.</p> <p>For Gateways running in a Batch Execution Environment, a unique IDPREFIX is created to enable the execution of all batch jobs with the same IDPREFIX value.</p> |
| ONLINE | Reserves the Gateway for online users only. This is the default for Gateways running in a CICS Execution Environment.   |

In the following table, a checkmark indicates a valid user type according to Execution Environment.

| Execution Environment | ANY       | ONLINE    | LOCAL |
|-----------------------|-----------|-----------|-------|
| Batch                 | Y (deflt) |           | Y     |
| Online                | Y         | Y         |       |
| CICS                  | Y         | Y (deflt) | Y     |

#### VERIFYEXTACCESS

Specifies how the Gateway should behave at Gateway initialization when one or more databases defined in the Gateway PSB are not available. For Gateways executing in a CICS Execution Environment, this parameter has meaning only if THREADUSAGE=SESSION. Valid entries are CALL and STARTUP. The default is CALL. Depending on the setting of this parameter and the type of Gateway, the following actions occur:

| VERIFYEXT<br>ACCESS | Gateway Type  | Action  |
|---------------------|---|---|
| CALL                | All   | Database access is determined at the time of the DL/I call.   |
| STARTUP             | Gateway using the DL/I or BMP interface with<br>THREADUSAGE=SESSION | When the Gateway starts, the status of any unavailable database appears in the Gateway job log. If the IMS status code NA or NU is encountered Gateway initialization fails, otherwise, initialization completes. |
| STARTUP             | Gateway using the CICS interface with<br>THREADUSAGE=SESSION        | When the Gateway starts, the status of any unavailable database appears in the Gateway job log, the PSB terminates, and Gateway initialization completes.   |



If this parameter is set to CALL and access to an unavailable database is requested, the ERROR exception is raised and the corresponding IMS status code (NA or NU) is passed back to TIBCO Object Service Broker.

### Estimating the CTABLESIZE Parameter

When you select IMS fields as TIBCO Object Service Broker IMS fields, the number of fields you can access using an IMS table definition is dependent upon the CTABLESIZE Data Object Broker parameter. To estimate the number of bytes required to support a specified number of fields, execute the following rule:

ESTIMATETBLDFN(*num\_fields*)

You must supply one argument, which is the maximum number of fields accessed by an IMS table in your system. The rule returns an estimate of the maximum CTABLESIZE required (for each TIBCO Object Service Broker table type) to support this number of fields.



Result of Executing ESTIMATETBLDFN for 50 Fields

```
----- INFORMATION LOG -----
COMMAND ===>
DATE: Nov 28,2006          REPORT ON ESTIMATE CTABLESIZE
                           FOR "50" FIELDS
                           SCROLL ===> P

Table Type      CTablesize(K)      XTablesize(K)
-----
ADA             5
DAT             7
DB2             5
IDM             6
IMS             6              3
MAP             4
SLK             4
TDS             3

PFKEYS: 2=NEXT LOG 3=EXIT 5=REPEAT FIND 12=EXIT 13=PRINT 9=RECALL
```

See Also     *TIBCO Object Service Broker for z/OS Installing and Operating* for more information on the CTABLESIZE parameter.

## Dynamically Changing Gateway Parameters

### Table Type Attributes

When a table is defined, attributes specific to external DBMS table types are held in the @SERVERPARMS TIBCO Object Service Broker control table, which is parameterized by table type. Each occurrence in the table specifies a value for the table about the external environment, such as SERVERID and DBID.

### Sample @SERVERPARMS Control Table for Table Type IMS

BROWSING TABLE : @SERVERPARMS(IMS)  
COMMAND ==>

| NUMBER | NAME           | TYPE | SYNTAX | LENGTH | DECIMAL | DEFAULT  | USAGE |
|--------|----------------|------|--------|--------|---------|----------|-------|
| 1      | SERVERID       | S    | C      | 8      | 0       | DEFAULT  | C     |
| 2      | SERVERTYPE     | S    | C      | 3      | 0       | IMS      | T     |
| 3      | DBNAME         | S    | C      | 8      | 0       | CUSTOMER | D     |
| 4      | PSBNAME        | S    | C      | 8      | 0       |          | P     |
| 5      | OPTIMIZEUPDATE | S    | C      | 1      | 0       | Y        | O     |

PFKEYS: 1=HELP 5=FIND NEXT 9=RECALL 18=EXCLUDE 13=PRINT 3=END 14=EXPAND

### Using SETXPARM and RESETXPARM

At runtime, you can dynamically modify the SERVERID gateway startup parameter using the utilities [SETXPARM](#) and [RESETXPARM](#). This reduces the number of table definitions required to define the external data. The changes are stored in either of two session tables:

|               |   |
|---------------|---|
| @SRVRPRMS_TYP | Manages global changes to the table type. |
|---------------|---|

**@SRVRPRMS\_TBL**

Manages specific changes to an individual table

The changes are in effect for the duration of the session, until [SETXPARM](#) is invoked again, or the overrides are reset.

## Gateway Parameters That Can Be Overridden at Runtime

The following Gateway parameters and Table Definer fields can be dynamically changed with [SETXPARM](#) and [RESETXPARM](#):

| Name           | Parameter or Field | Default Value | Maximum Length (Bytes) |
|----------------|--------------------|---------------|------------------------|
| SERVERID       | Parameter          | DEFAULT       | 8                      |
| PSBNAME        | Parameter          |               | 8                      |
| SERVERTYPE     | Parameter          | IMS           | 3                      |
| DBNAME         | Field              |               | 8                      |
| OPTIMIZEUPDATE | Field              | N             | 1                      |

## Examples Using SETXPARM and RESETXPARM

This example sets the SERVERID for all IMS tables to TORONTO.

```
CALL SETXPARM('TABLETYPE', 'IMS', 'SERVERID', 'TORONTO', '');
```

This example sets the value of the **Optimize Update** field for the EMPLOYEE table to Y:

```
CALL SETXPARM('TABLENAME', 'EMPLOYEE', 'OPTIMIZEUPDATE', 'Y', '');
```

This example resets the SERVERID for IMS tables to the Table Definer default value.

```
CALL RESETXPARM ('TABLETYPE', 'IMS', 'SERVERID', '');
```

See Also *TIBCO Object Service Broker Shareable Tools* for detailed descriptions of the [SETXPARM](#) and [RESETXPARM](#) tools.

## Implementing External Security

---

External security is used only if SECLEVEL=1. This security is in addition to any security done when SECLEVEL=0. The external security interface verifies that a TIBCO Object Service Broker authorization ID has access to specific IMS resources at the TIBCO Object Service Broker transaction level.

The Gateway issues SAF RACROUTE calls to a SAF-compliant external security package such as CA-ACF2, RACF, or CA-Top Secret to verify the access to specific IMS resources. Refer to the *External Security Interface (RACROUTE) Macro Reference* IBM manual for z/OS and VM for more information. You must define the appropriate IMS resources and the allowable accesses (read, update, and so on) in the external security package you are using. You must perform this for each existing TIBCO Object Service Broker authorization ID that accesses IMS data using a Gateway with SECLEVEL=1.

### Security Levels

The external security interface provides six combinations of security levels to check IMS resources. You can choose one of the following methods:

- Resource
- PSBname
- Database
- Segment
- Resource and Database
- Resource and Segment

The external security interface is invoked at the start of a TIBCO Object Service Broker transaction for the specified authorization ID. The EXTERNALUSERID and EXTERNALGROUP parameters determine whether the authorization ID is the TIBCO Object Service Broker session ID, the current security group, or a combination of the two.

The Gateway builds a profile in its buffers for each initial SAF request for a TIBCO Object Service Broker authorization ID. The next time an authorization ID requests access to an IMS resource, the Gateway checks its buffers to determine if a profile corresponding to this authorization ID exists. If there is a corresponding profile, the Gateway uses it instead of SAF to verify access to the specified IMS resource.

## Information Logged in TIBCO Object Service Broker Authorization ID Profile

The information logged in the profile and stored in the Gateway buffers for each TIBCO Object Service Broker authorization ID depends upon the level of security requested as described in the table below:

| Security Level | Logged Information                                  |
|----------------|---|
| Resource       | Successful accesses to the RESOURCE parameter.      |
| PSBname        | Successful accesses to the PSB.                     |
| Database       | Successful accesses to a database.                  |
| Segment        | Successful accesses to a segment within a database. |



Since the Gateway logs only successful accesses, you must recycle the Gateway to block access. You do not have to recycle the Gateway to grant access.

## Authorizing the Gateway for SAF

The Gateway must run APF authorized so that it can issue SAF calls. Therefore, the STEPLIB must be APF authorized.

## Establishing Resource Security

To establish resource-level security, specify the gateway parameters as described in the table below:

| Parameter | Value Required  |
|-----------|---|
| SECLEVEL  | 1   |
| CLASS     | SAF class name to which the resource belongs.   |
| RESOURCE  | Resource name.  |
| DECOUPL   | YES for RACF.<br>NO for any other security package. Review the SUBSYS and REQSTOR parameters and modify them as required. |

The Gateway issues SAF calls to verify that the TIBCO Object Service Broker authorization ID has access to the IMS resource within the specified SAF class. Refer to [Establishing the TIBCO Object Service Broker Authorization ID on page 82](#) and [Supplying Gateway Startup Parameters on page 57](#) for more information.

The external security package must have the SAF interface activated and the following items defined:

- SAF classes
- Resources within each SAF class
- Read and update access allowed on each resource for each TIBCO Object Service Broker authorization ID



Security for resource is checked once at transaction start.

### Establishing PSB Security

To establish PSB-level security, specify the gateway parameters as described in the table below:

| Parameter | Value Required  |
|-----------|---|
| SECLEVEL  | 1   |
| PSBCLASS  | SAF class name of the IMS PSBs.   |
| DECOUPL   | YES for RACF.<br>NO for any other security package. Review the SUBSYS and REQSTOR parameters and modify them as required. |

The Gateway issues SAF calls to verify that the TIBCO Object Service Broker authorization ID has access to the IMS PSB within the specified SAF class. Refer to [Establishing the TIBCO Object Service Broker Authorization ID on page 82](#) and [Supplying Gateway Startup Parameters on page 57](#) for more information.

The external security package must have the SAF interface activated and the following items defined:

- SAF classes
- PSB within each SAF class
- Read and update access allowed on each resource for each TIBCO Object Service Broker authorization ID.



Security for PSBs is checked once per transaction.

## Establishing Database Security

To establish database-level security, specify the gateway parameters as described in the table below:

| Parameter | Value Required  |
|-----------|---|
| SECLEVEL  | 1   |
| DBDCLASS  | SAF class name of the IMS databases.  |
| DECOUPL   | YES for RACF.<br>NO for any other security package. Review the SUBSYS and REQSTOR parameters and modify them as required. |

The Gateway issues SAF calls to verify that the TIBCO Object Service Broker authorization ID has access to the database specified in the IMS table definition within the specified SAF DBDCLASS. Refer to [Establishing the TIBCO Object Service Broker Authorization ID on page 82](#) and [Supplying Gateway Startup Parameters on page 57](#) for more information.

The external security package must have the SAF interface activated and the following items defined:

- SAF database class
- All IMS databases defined as resources within the SAF database class
- Read and update access allowed on each resource (IMS databases) for each TIBCO Object Service Broker authorization ID

## Establishing Segment Security

To establish segment-level security, specify the gateway parameters as described in the table below:

| Parameter | Value Required  |
|-----------|---|
| SECLEVEL  | 1   |
| SEGCLASS  | SAF class name of the IMS segments.   |
| DECOUPL   | YES for RACF.<br><br>NO for any other security package. Review the SUBSYS and REQSTOR parameters and modify them as required. |

The Gateway issues SAF calls to verify that the TIBCO Object Service Broker authorization ID has access to the database and segments specified in the IMS table definition within the specified SAF SEGCLASS. Refer to [Establishing the TIBCO Object Service Broker Authorization ID on page 82](#) and [Supplying Gateway Startup Parameters on page 57](#) for more information.

The external security package must have the SAF interface activated and the following items defined:

- SAF segment class
- All IMS segments of each database defined as resources within the SAF segment class
- Read and update access allowed on each resource (IMS segments within databases) for each TIBCO Object Service Broker authorization ID

## Establishing Resource and Database Security

To establish resource- and database-level security, specify the gateway parameters as described in the table below:

| Parameter | Value Required  |
|-----------|-----------------|
| SECLEVEL  | 1               |
| CLASS     | SAF class name. |
| RESOURCE  | Resource name.  |



| Parameter | Value Required  |
|-----------|---|
| DBDCLASS  | SAF class name of the IMS databases.  |
| DECOUPL   | YES for RACF.<br><br>NO for any other security package. Review the SUBSYS and REQSTOR parameters and modify them as required. |

The Gateway issues SAF calls to verify the TIBCO Object Service Broker authorization ID has access to the IMS resource in the specified SAF class. The Gateway issues another SAF call to verify the authorization ID has access to the database specified in the IMS table definition in the specified SAF DBDCLASS. Refer to [Establishing the TIBCO Object Service Broker Authorization ID on page 82](#) and [Supplying Gateway Startup Parameters on page 57](#) for more information.

The external security package must have the SAF interface activated and the following items defined:

- SAF classes
- Resources within each SAF class
- SAF database class
- All IMS databases defined as resources within the SAF database class
- Read and update access allowed on each resource (IMS databases and all other IMS resources) for each authorization ID

## Establishing Resource and Segment Security

To establish resource- and segment-level security specify the gateway parameters as described in the table below:

| Parameter | Value Required   |
|-----------|--|
| SECLEVEL  | 1  |
| CLASS     | SAF class name.  |
| RESOURCE  | Resource name.   |
| SEGCLASS  | SAF class name of IMS databases and segments in each database. |

| Parameter | Value Required  |
|-----------|---|
| PSBNAME   | The Gateway PSB.  |
| DECOUPL   | YES for RACF.<br><br>NO for any other security package. Review the SUBSYS and REQSTOR parameters and modify them as required. |

The Gateway issues SAF calls to verify the TIBCO Object Service Broker authorization ID has access to the IMS resource in the specified SAF class. The Gateway issues another SAF call to verify that the authorization ID has access to the database and segments specified in the IMS table definition within the specified SAF segment class. Refer to [Establishing the TIBCO Object Service Broker Authorization ID on page 82](#) and [Supplying Gateway Startup Parameters on page 57](#) for more information.

The external security package must have the SAF interface activated and the following items defined:

- SAF classes
- Resources within each SAF class
- SAF segment class
- All IMS database and segment combinations defined as resources within the SAF segment class
- Read and update access allowed on each resource (all IMS database and segment combinations and all other IMS resources) for each authorization ID

### Establishing the TIBCO Object Service Broker Authorization ID

The Gateway uses the EXTERNALUSERID and EXTERNALGROUP startup parameters to build the authorization ID. SAF uses this ID to verify access to IMS resources. The table below illustrates how TIBCO Object Service Broker evaluates the values you specify for these parameters:

## Userid and Group ID Values

| Gateway Parameter | Value         | Authorization ID passed to SAF  |
|-------------------|---------------|---|
| SECLEVEL          | 1             | SAF user ID = TIBCO Object<br>Service Broker session ID   |
| EXTERNALUSERID    | Not specified |   |
| EXTERNALGROUP     | Not specified |   |
| SECLEVEL          | 1             | SAF user ID = TIBCO Object<br>Service Broker session ID   |
| EXTERNALUSERID    | USERID        |   |
| EXTERNALGROUP     | not specified |   |
| SECLEVEL          | 1             | SAF user ID = TIBCO Object<br>Service Broker session ID<br>SAF group ID = name of current<br>security group. If the name is<br>longer than 8 characters, the SAF<br>group name is set to blank. |
| EXTERNALUSERID    | USERID        |   |
| EXTERNALGROUP     | GROUP         |   |
| SECLEVEL          | 1             | SAF user ID = name of current<br>security group. If current security<br>group name is more than 8<br>characters, a SECURITYFAIL is<br>signalled.  |
| EXTERNALUSERID    | GROUP         |   |
| EXTERNALGROUP     | n/a           |   |

## Implementing Fail Safe Processing

---

Fail Safe level-1 processing guarantees consistency when updating both TDS and IMS data from a single instance of the Gateway in a single transaction. At the end of a transaction, the Data Object Broker requests that the Gateway commit outstanding updates. As part of IMS commit processing, the Gateway updates an IMS transaction database to record the successful commit. If the Gateway does not respond to the Data Object Broker in a reasonable amount of time, the transaction is flagged as being in doubt. Locks held on TDS data remain in place until the problem is resolved.

When a connection is re-established between the Data Object Broker and a Gateway with the same configuration as the one that failed, the Data Object Broker asks the Gateway if the in-doubt transaction completed. The Gateway checks the IMS transaction database to determine this. If the update is completed in IMS, the TDS updates are applied and the locks are released.



For the concurrent access to DB2 data, DB2 Fail Safe processing is used.



You can resolve only in-doubt transactions by starting an instance of Gateway with exactly the same parameter settings as the Gateway in use at the time the transaction was placed in doubt.

See Also

*TIBCO Object Service Broker for z/OS Managing Backup and Recovery* for more information on Fail Safe processing.

## Procedural Overview

To implement Fail Safe level-1 processing, you must:

1. [Define an IMS transaction database, page 85](#)
2. [Define a PCB for the transaction database in each Gateway PSB, page 85](#)
3. [Define a TIBCO Object Service Broker IMS transaction table, page 85](#)
4. [Modify the Gateway Fail Safe startup parameters, page 85](#)

These tasks are described in the following sections.

### Task A Define an IMS transaction database

Member XIMSTRXD exists in the CNTL data set. Modify this member as explained in the USAGE section at the top of the member. The transaction database holds a maximum of one segment occurrence for each combination of the Gateway and a Data Object Broker.

### Task B Define a PCB for the transaction database in each Gateway PSB

Member XIMSTRXP exists in the CNTL data set. Modify this member as explained in the USAGE section at the top of the member.

### Task C Define a TIBCO Object Service Broker IMS transaction table

After defining an IMS transaction database, you must extract and load the transaction database into TIBCO Object Service Broker. When this is completed, you must define a TIBCO Object Service Broker IMS transaction table that points to the IMS transaction database. The default name for this table is @IMSFSTRXDB.

### Task D Modify the Gateway Fail Safe startup parameters

Ensure that the following Gateway startup parameters are included in your Gateway startup JCL:

- FSLEVEL=1
- Optionally, one or both of:
  - FSTABLENAME=@IMSFSTRXDB
  - TRXDB=IMS.DBname

For more information on these Gateway startup parameters, refer to [Supplying Gateway Startup Parameters on page 57](#).

At startup, the Gateway asks the Data Object Broker for the TIBCO Object Service Broker IMS transaction table definition specified in the FSTABLENAME gateway startup parameter. This table definition is bound for the life of the Gateway and is used at transaction end to update the IMS transaction ID database.

The transaction table can be managed in TIBCO Object Service Broker like any other IMS table. For example, you can write a TIBCO Object Service Broker rule to clean up the Fail Safe database when shutting down the Gateway.



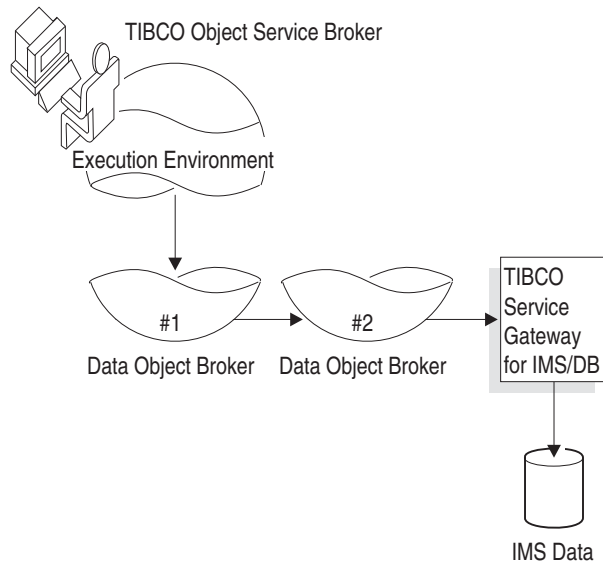
The default name of the TIBCO Object Service Broker IMS transaction table (@IMSFSTRXDB) could have been modified when you defined it in [Define a TIBCO Object Service Broker IMS transaction table on page 85](#).

## Understanding Other Operational Procedures

### Using Distributed Data with Service Gateway for IMS/DB

Distributed access between TIBCO Object Service Broker and IMS is permitted subject to requirements of all distributed access. Refer to *TIBCO Object Service Broker Application Administration* and *TIBCO Object Service Broker for z/OS Installing and Operating* for more information on distributed access.

This illustration shows a sample TIBCO Object Service Broker IMS distributed-data scenario:



### Displaying the Status of the Gateway

Use the **RESOURCE MANAGEMENT** option from the Administration menu to display the status of the Gateway. The example below shows the type of information displayed for **SERVER TYPE=IMS**.

Sample Resource Detail

|  |              |                                  |     |           |        |        |                    |       |      |     |
|--|--------------|----------------------------------|-----|-----------|--------|--------|--------------------|-------|------|-----|
| S6BADM33   | HTSTSRV      | RESOURCE DETAIL FOR IMS IMSDRASV |     |           |        |        | 2000JAN20 16:40:52 |       |      |     |
| INTERMEDIATE ROLLBK  | N            | EARLY RELEASE                    | Y   | LAST USER | REUSE  | N      | COMMIT LEVEL       | 0     |      |     |
| RETRY INTERVAL   | 0            | TP NAME                          |     | USER ID   | PREFIX |        | FAILURES           | 0     |      |     |
| NODE   | MMMSYSTEMA01 | INDOUBTS                         | N   |           |        |        | DELETE             |       |      |     |
|  |              | CONNECTIONS                      |     | IN-USE    |        | TRX    | MESSAGE            |       |      |     |
|  | CUR          | MAX                              | LMT | CUR       | MAX    | COUNT  | COUNT              |       |      |     |
| ONLINE   | 0            | 0                                | 0   | 0         | 0      | 0      | 0                  |       |      |     |
| COMMON   | 0            | 0                                | 25  | 0         | 0      | 0      | 0                  |       |      |     |
| SCHEDULE NAME IMSIMSDRASV  |              |                                  |     |           |        |        |                    |       |      |     |
| APPLICABLE DAYS  |              |                                  |     | EXCEPTION | START  | ONLINE | CONNECTIONS        |       |      |     |
| MON  | TUE          | WED                              | THR | FRI       | SAT    | SUN    | DATE               | TIME  | ONLY | MAX |
| Y  | Y            | Y                                | Y   | Y         | Y      |        |                    | 00:00 | N    | 25  |
| ENTER-PATHS PF2-TYPE PF4-GROUP PF5-PEER PF9-START PF10-SCHEDULES PF11-UPDATE |              |                                  |     |           |        |        |                    |       |      |     |
| THERE ARE NO ACTIVE PATHS TO BE DISPLAYED, REQUEST IGNORED                   |              |                                  |     |           |        |        |                    |       |      |     |

Defining User Exits with IMSUSRXT

Modify the ASM data set member IMSUSRXT to define your own IMS user exits. Use IMSUSRXT by calling it from the Gateway before issuing DL/I calls.

A typical use for IMSUSRXT is to alter the SSA used by the Gateway. For example, you can change the command code from \*D to \*L such that you can retrieve the last segment and read the database backwards.

On entry to IMSUSRXT, register 0 points to the TIBCO Object Service Broker table name:

|          |                   |      |            |
|----------|-------------------|------|------------|
| *****    |                   |      |            |
| *        | TABLE INFORMATION |      | *          |
| *****    |                   |      |            |
| CTABLE   | DSECT             | ,    |            |
|          | DS                | XL97 |            |
| CTABNAME | DS                | CL16 | TABLE NAME |

and the register contains:

```
LR      R8,R0
USING   CTABLE,R8
CLC      CTABNAME,=Cyour_table_name
BNE      EXIT00
```

Register 1 points to the DL/I PARM LIST:

```
*****
*          DLI PARM LIST                      *
*****
DLILIST  DSECT          IMS DL/I CALL PARAMETERS
DLIFUNC  DS      A      FUNCTION VECTOR
DLIPCB   DS      A      PCB VECTOR
DLIAREA  DS      A      I/O AREA VECTOR
DLISSA   DS      15F     SSA VECTORS
DLILSTLN EQU  *-DLILIST  DL/I CALL LIST LENGTH
```

and the register contains:

```
LR      R4,R1
USING   DLILIST,R4
L       R5,DLISSA      ADDRESS OF SSA VECTORS
L       R6,0(,R5)      FIRST SSA
```



To invoke your user exit, add INCLUDE(IMSUSRXT) to the JCL member XLINKIMS in module S6BIMS. For example:

---

```

ORDER    COPYRGHT
INCLUDE  RESLIB(ASMTDLI)
INCLUDE  RESLIB(DFSLI000)
INCLUDE  SCEELKED(CEEBETBL)
INCLUDE  SCEELKED(CEEBINT)
INCLUDE  SCEELKED(CEEBLLST)
INCLUDE  SCEELKED(CEEBPUBT)
INCLUDE  SCEELKED(CEEBTRM)
INCLUDE  SCEELKED(CEESG003)
INCLUDE  SCEELKED(CEESTART)
INCLUDE  ALOAD(CLKXMSTR)
INCLUDE  ALOAD(IMSCALLD)
INCLUDE  ALOAD(IMSCALLS)
INCLUDE  ALOAD(IMSDB2M$)
INCLUDE  ALOAD(IMSDTO)
INCLUDE  ALOAD(IMSDREQ)
INCLUDE  ALOAD(IMSLEAS)
INCLUDE  ALOAD(IMSPPSB)
INCLUDE  ALOAD(IMSREC)
INCLUDE  ALOAD(IMSROW)
INCLUDE  ALOAD(IMSSEC)
INCLUDE  ALOAD(IMSSREQ)
INCLUDE  ALOAD(IMSSRVR)
INCLUDE  ALOAD(IMSSSA)
INCLUDE  ALOAD(MSGEDIT)
INCLUDE  ALOAD(PRODUCT)
INCLUDE  ALOAD(SRVMSG)
INCLUDE  ALOAD(SRVSAFX)
INCLUDE  your_object_lib(IMSUSRXT)
SETCODE  AC(1)
MODE     AMODE(31),RMODE(24)
ENTRY    IMSSRVR
ALIAS    HRNIM2
ALIAS    HRNIMS
NAME     S6BIMS(R)

```

---

### Sample IMS User Exit IMSUSRXT

A sample IMS user exit is included as member IMSUSRXT in the ASM data set. Modify it as required.

## Debugging Information

You can use the TIBCO Object Service Broker Rule Debugger to identify and fix errors in your applications. You can also make and test certain ad hoc changes to your rules. The Debugger stops the running of the rule at events you specified in the Debug screen.



For concurrent DB2 accesses, you must determine the SQL statements being sent to DB2 by your rules. Refer to [Supplying Gateway Startup Parameters on page 57](#) for information on the TRACE parameter.

### Using the TRACE and DEBUG Parameters

For the Gateway to determine the SSAs built and sent, include the TRACE parameter in the IMSSRV00 SYSIN file. This requires a dedicated Gateway since the SSAs for all IMS accesses are included in the trace. This can be accomplished using the SERVERID parameter. To change the server ID of an IMS table definition to an alternate dedicated server ID, execute the [CHANGE\\_SERVERID](#) tool as follows:

**CHANGE\_SERVERID**(*table\_name*,*old\_serverid*,*new\_serverid*)

The trace output also includes information similar to the DL/I Trace Facility. For more information on gateway parameters, refer to [Supplying Gateway Startup Parameters on page 57](#).

In addition to the trace output, you can also use the DEBUG gateway parameter to echo all requests received by the Gateway to the JES job log for batch jobs or to the console for started tasks. For more information refer to [Supplying Gateway Startup Parameters on page 57](#).

### Using the DL/I Trace Facility

If you are running an instance of the Gateway using the DL/I or DBB interface, you can use the DL/I Trace Facility to assist in debugging problems. To trace DL/I calls created by the Gateway, make the following modifications:

- Add the DFSTROUT DD statement in the Gateway startup JCL:  
//DFSTROUT DD SYSOUT=\*,DCB=(LRECL=80,RECFM=B,BLKSIZE=8000)
- Add the following line to the DFSVSAMP data set:  
DLITRACE LOG=NO,COMP

To view only the DL/I statements created by your rules, you require a dedicated Gateway. This can be accomplished using the SERVERID parameter. To change the server ID of an IMS table definition to an alternate dedicated server ID, execute the [CHANGE\\_SERVERID](#) tool as follows:

**CHANGE\_SERVERID**(*table\_name*,*old\_serverid*,*new\_serverid*)

In addition to the DL/I Trace Facility, you can also use the DEBUG gateway parameter to echo all requests received by the Gateway to the JES job log for batch jobs, or to the console for started tasks. For more information, refer to the DEBUG parameter in [Supplying Gateway Startup Parameters on page 57](#).



Use the TRACE and DEBUG parameters *only* for error resolution or development. The additional overhead required to log information can degrade performance.

## Reporting Problems

Refer to [How to Contact TIBCO Support on page xvi](#) to report problems to TIBCO Support. Have the following information available:

- DBD listing
- PSB listing
- DL/I Trace output
- DL/I Gateway trace output
- Gateway JCL and job log
- Data Object Broker job log
- IMS table definitions and sample data
- Listings of the following TIBCO Object Service Broker IMS control tables:
  - @IMSCONTROL(*table*)
  - @IMSDBD(*dbname, segname*)
  - @IMSDBS (section containing tables in question)
  - @IMSFIELDS(*table*)
  - @IMSSEGS(*dbname*)
  - @IMSSIXFIELDS(*logical dbname, logical target segname, logical fieldname*)
  - @IMSTABLES (section containing tables in question)
  - @SERVERPARMS(*table*).

**See Also** *TIBCO Object Service Broker Application Administration* and *TIBCO Object Service Broker for z/OS Installing and Operating* for more information on distributed access.

*TIBCO Object Service Broker for z/OS Installing and Operating* for more information on the RESOURCE MANAGEMENT option.

*TIBCO Object Service Broker Programming in Rules* for more information on debugging rules and applications.

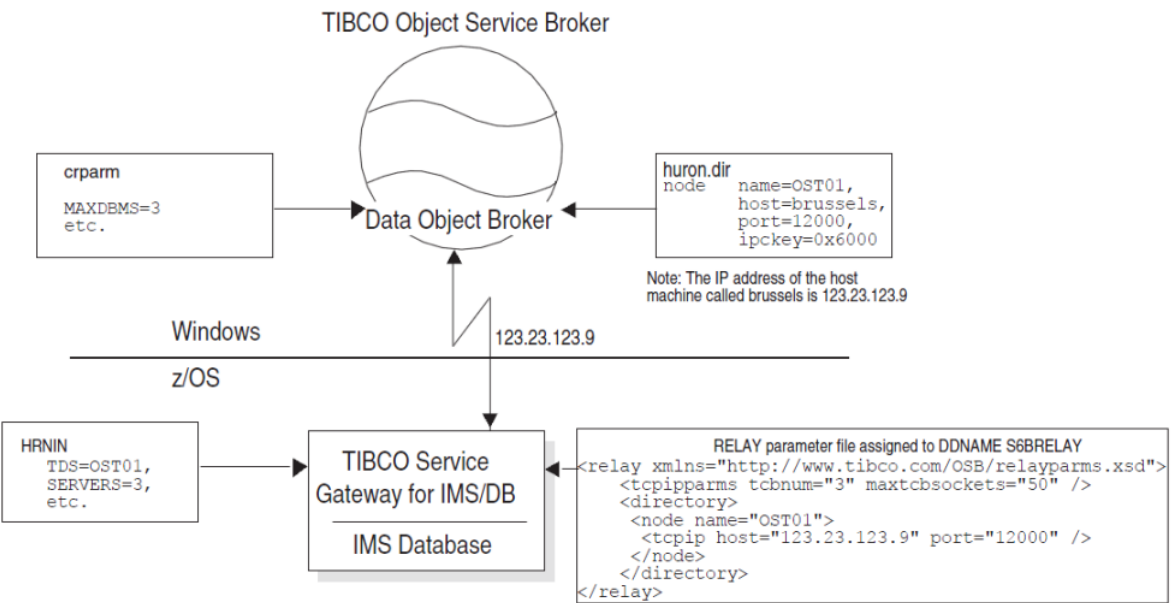
# Connecting the Gateway to a Windows or Solaris Data Object Broker

You can configure the Data Object Broker and the Gateway to reside on different domains and operating systems (z/OS, Windows, or Solaris). The Gateway must be in the same domain as the IMS database system. The following configuration steps are required to access a Data Object Broker from a different operating environment than your Gateway:

- Configure the TCP/IP connection on the z/OS system where your Gateway and IMS database reside.
- Configure the TCP/IP connection on the machine where your TIBCO Object Service Broker for Open Systems resides.
- Specify the number of Gateways that can connect to the Data Object Broker.
- Specify the appropriate Gateway parameters.

## Sample Configuration

The following diagram shows a sample configuration:



## Configure the TCP/IP Connection on z/OS

Prepare the TIBCO Object Service Broker relay file (RELAYCFG member in the CNTL data set). This file associates the TIBCO Object Service Broker communications identifier with the TCP/IP application addressing information.

### Sample Relay File Assigned to DDNAME S6BRELAY

```
<relay xmlns="http://www.tibco.com/OSB/relayparms.xsd">
  <tcpipparms tcbnum="3" maxtcbsockets="50" />
  <directory>
    <node name="OST01">
      <tcpip host="123.23.123.9" port="12000" />
    </node>
  </directory>
</relay>
```



The element and attribute names in the relay file are case sensitive.

See Also

*TIBCO Object Service Broker for z/OS Installing and Operating* for detailed information about preparing the TIBCO Object Service Broker relay file.

*TIBCO Object Service Broker Parameters* for details about the parameters used in the relay file and how to specify them.

## Configure the TIBCO Object Service Broker TCP/IP Environment

Add the following parameters for the TCP/IP connection to the Data Object Broker directory file, *huron.dir*:

|        |  |
|--------|--|
| name   | This must be the same value as the node name set in the relay file described in <a href="#">Configure the TCP/IP Connection on z/OS on page 93</a> . |
| host   | The name of the host machine where the TIBCO Object Service Broker monitor process listens for connections.  |
| port   | The number of the TIBCO Object Service Broker monitor socket port.   |
| ipckey | The value of the IPC key.  |

## Specify the Number of Gateways Connecting to the Data Object Broker

Specify the following value for the MAXDBMS parameter in the crparm file for your Data Object Broker.

---

|         |   |
|---------|---|
| MAXDBMS | This must be equal to or greater than the value specified in the SERVERS= parameter HRNIN DD statement for startup. |
|---------|---|

---

## Specify the Gateway Parameter

Specify the following value for the TDS= parameter HRNIN DD statement for startup. Refer to [Supplying Gateway Startup Parameters on page 57](#) for more details about specifying gateway parameters.

---

|     |  |
|-----|--|
| TDS | This must be the same value as the node name set in the relay file described in <a href="#">Configure the TCP/IP Connection on z/OS on page 93</a> . |
|-----|--|

---

See Also *TIBCO Object Service Broker for z/OS Installing and Operating* for detailed information about preparing the relay file.

*TIBCO Object Service Broker Parameters* for details about the parameters used in the relay file and how to specify them.

## Chapter 3      **Managing IMS Data Definitions**

This chapter elaborates on how to manage TIBCO Object Service Broker IMS Data Definitions and how to use the TIBCO Object Service Broker IMS table.

### Topics

---

- [Managing IMS Data from TIBCO Object Service Broker, page 96](#)
- [Task A: Extract the IMS Database Definition, page 96](#)
- [Task B: Invoke the Table Definer, page 97](#)
- [Task C: Specify Header Information, page 101](#)
- [Task D: Select an IMS Database, page 102](#)
- [Task E: Define a Location Parameter, page 106](#)
- [Task F: Select IMS Segments, page 107](#)
- [Task G: Select IMS Fields, page 109](#)
- [Task H: Define Event Rule Information, page 117](#)
- [Defining Secondary Indexes, page 119](#)
- [Managing IMS Data from TIBCO Object Service Broker, page 96](#)

## Managing IMS Data from TIBCO Object Service Broker

---

To access IMS data from TIBCO Object Service Broker, you must define a TIBCO Object Service Broker table of type IMS. A TIBCO Object Service Broker IMS table can have up to four data parameters, up to five fields as primary keys, one or more additional fields, and an optional location parameter.

Complete the following tasks to define an IMS table:

1. [Task A: Extract the IMS Database Definition, page 96](#)
2. [Task B: Invoke the Table Definer, page 97](#)
3. [Task C: Specify Header Information, page 101](#)
4. [Task E: Define a Location Parameter, page 106](#)
5. [Task F: Select IMS Segments, page 107](#)
6. [Task F: Select IMS Segments, page 107](#)
7. [Task G: Select IMS Fields, page 109](#)
8. [Task H: Define Event Rule Information, page 117](#)

These tasks are defined in the sections below.

### Task A: Extract the IMS Database Definition

---

Before you can define an IMS table, you must describe the IMS database to TIBCO Object Service Broker. Refer to [Task A, Extract IMS database information, on page 50](#) for more information.



## Task B: Invoke the Table Definer

---

Invoke the Table Definer from the workbench using the DT define table option. You can modify an existing definition or define a new IMS table.

### Accessing Existing Tables

You can display the definition of an existing IMS table from the workbench in one of three ways:

- Type the name of an existing table beside the DT define table option and press Enter to display its definition.
- Type the name of an existing table in the primary command field, for example:  
`DT CUST_ORDERS<Enter>`
- Move the cursor to the DT define table option and press Enter. This displays the Object Manager screen, which contains a list of existing tables.

Scroll through this list to see which table you require. To select a table, type **s** beside the name and press Enter.

### Defining a New Table

To define a new table, complete the following steps:

1. Type the name of a new IMS table beside the DT define table option or in the primary command field and press Enter.  
A TDS definition template appears.

- 2. Change the Type field at the top of the screen from TDS to IMS and press Enter.

A Table Definition screen similar to the one shown below appears.

COMMAND==>

TABLE DEFINITION

Table: CUST\_ORDERS

Type: IMS

Unit: USR40

DB Name:

Server ID: DEFAULT

Server Type: IMS

Optimize Update:

PSBname: OSTRXDRP

Browse PSBname: OSTRXDRQ

Location Parm

Typ

Syn

Len

Dec

Default

Src

Sourcename

LOCATION

I

C

16

0

Level

Segment Name

Fields Chosen

Order

From Segment

0

Event Rule

Type

Access

Parms Chosen

Order

From Segment

PFKEYS: 3=SAVE 12=CANCEL 22=DELETE 13=PRINT 6=CHILD SEGMENTS 2=DOC 4=DB NAMES

### IMS Table Definition Screen Segments

The IMS table definition screen contains the following segments:

|                            |  |
|----------------------------|--|
| Header segment             | Specifies the name, type, and unit specifications for the table. Refer to <a href="#">Task C: Specify Header Information on page 101</a> for more information.       |
| IMS Database segment       | Specifies the IMS database on which you want to base the table definition. Refer to <a href="#">Task D: Select an IMS Database on page 102</a> for more information. |
| Location Parameter segment | Specifies a location parameter for the IMS table. Refer to <a href="#">Task E: Define a Location Parameter on page 106</a> for more information.                     |

|                    |  |
|--------------------|--|
| IMS segment        | Displays the IMS segments on which you want to base your IMS table. Refer to <a href="#">Task F: Select IMS Segments on page 107</a> for more information. |
| Fields segment     | Displays the IMS fields you chose from the selected IMS segments. Refer to <a href="#">Task G: Select IMS Fields on page 109</a> for more information.     |
| Event Rule segment | Specifies event rules for the IMS table. Refer to <a href="#">Task H: Define Event Rule Information on page 117</a> for more information.                  |
| Parameter segment  | Displays data parameters for the IMS table. Refer to <a href="#">Task G: Select IMS Fields on page 109</a> for more information.                           |
| PF Keys segment    | Displays the PF keys available from this screen. Refer to the table below for more information.  |

## PF Keys and Primary Commands

Use the following PF keys or primary commands (or their abbreviations) from the IMS Table Definition screen:

| PF Key | Primary Command | Description  |
|--------|-----------------|--|
|        | <b>COPY</b>     | Copies the definition of an existing IMS table into the current definition.  |
| PF1    | -               | Displays corresponding help for the current field or screen.   |
| PF2    | <b>DOC</b>      | Displays the documentation screen, where you can document the table definition. Refer to <a href="#">Appendix A, Documenting IMS Tables, on page 141</a> for more information. |
| PF3    | <b>SAVE</b>     | Saves the existing definition and returns you to the workbench.  |
| PF4    | <b>DB NAMES</b> | Lists existing IMS databases upon which you can base your IMS table.   |
| PF6    | -               | Displays the child segments of the current segment.  |

| PF Key | Primary Command | Description  |
|--------|-----------------|--|
| PF12   | <b>CANCEL</b>   | Cancels any changes to the definition and returns you to the workbench.            |
| PF13   | <b>PRINT</b>    | Prints the definition of the table. You remain in the Table Definer.               |
| PF22   | <b>DELETE</b>   | Deletes the definition of the IMS table. You are prompted to confirm the deletion. |

See Also *TIBCO Object Service Broker Getting Started* for more information on the Object Manager.

## Task C: Specify Header Information

---

### Table Definition Screen Fields

The Header segment of the IMS Table Definition screen contains the following fields:

---

|              |   |
|--------------|---|
| <b>Table</b> | <p>Displays the table name specified when you invoked the Table Definer. You can type in a new name to save the definition of the current table under a new name.</p> <p>Valid entries include a character string of up to 16 characters beginning with a letter (A-Z) or a special character (\$ or #), and continuing with more letters, special characters, digits (0-9), or underscore characters (_).</p> <p>A table name starting with an @ symbol denotes a table supplied by TIBCO Object Service Broker.</p> |
| <b>Type</b>  | <p>Displays the table type IMS, which you changed in <a href="#">Task B: Invoke the Table Definer on page 97</a>.</p>   |
| <b>Unit</b>  | <p>Displays the user unit associated with the table. The unit marks the table as belonging to a particular application or to a logical unit such as utilities, accounting, or network control. The default unit for your user ID is specified in your user profile.</p> <p>Valid entries include a character string of up to a maximum of eight characters. These can be provided by your system administrator, for example, ACC.</p>   |

---

Press Enter to validate the information you entered in the Header segment.

**See Also** *TIBCO Object Service Broker Shareable Tools* for more information on tools that you can use to copy TIBCO Object Service Broker objects.



Selecting an IMS Database

```
COMMAND==>                                TABLE DEFINITION

      Table: CUST_ORDERS      Type: IMS      Unit: USR40
      DB Name:
      Server ID: DEFAULT      Server Type: IMS  Optimize Update:
      PSBname: OSTRXDRP      Browse PSBname: OSTRXDRQ

      Location Parm      Typ Syn Len Dec Default      Src Sourcename
      -----
      _ LOCATION              I   C  16  0

      Level      Segment Name      Fields Chosen      Order From Segment
      --      -----
      _ 0          CUSTROOT

      Event Rule      Type Access      Parms Chosen      Order From Segment
      -----
      _
      _
      _
PFKEYS: 3=SAVE 12=CANCEL 22=DELETE 13=PRINT 6=CHILD SEGMENTS 2=DOC 4=DB NAMES
```

After you select an IMS database, complete the following fields to continue defining the IMS table definition:

|             |  |
|-------------|--|
| Server ID   | <p>Type the ID for a Gateway or group of Gateways to use when accessing the table you are defining. This ID identifies Gateways with common characteristics and must match the SERVERID startup parameter specified in the Gateway JCL (described in <a href="#">Supplying Gateway Startup Parameters on page 57</a>). Valid entries are character strings of up to eight characters. The default is DEFAULT.</p> <p>The SERVERID parameter can be overridden at runtime. Refer to <a href="#">Chapter 2, Operating the Service Gateway for IMS/DB, on page 49</a> for more information.</p> |
| Server Type | <p>Specifies access to IMS data or both IMS and DB2 data in the same TIBCO Object Service Broker transaction. Valid entries are:</p> <p><b>IMS</b> – Use the Gateway to access only IMS data.</p> <p><b>IM2</b> – Use the Gateway to access both IMS and DB2 data.</p>   |

---

|                        |   |
|------------------------|---|
| <b>Optimize Update</b> | Required only when one or more IMS segments defined in an IMS table do not have unique keys and the definition is used in an INSERT, REPLACE, or DELETE statement. The default is N. Valid entries: |
|------------------------|---|

---

---

Y The Gateway returns one segment occurrence at a time to the Data Object Broker. The following applies:

You cannot specify ordering in the IMS table definition or from rules.

All fields belonging to the segment where the primary key is defined must be fully defined to TIBCO Object Service Broker. If the definition contains more than one segment type under the TIBCO Object Service Broker primary key, all fields belonging to each segment type must be fully defined to TIBCO Object Service Broker.

The Gateway Program Specification Block (PSB) requires at least two identical Program Control Blocks (PCBs) for the specified IMS database. Refer to [Chapter 2, Operating the Service Gateway for IMS/DB, on page 49](#) for more information.

---

N For a FORALL statement, the Gateway returns IMS data to the Data Object Broker in variable length buffers up to 31 KB. If the request must have more than 31 KB of data, it sends 31 KB buffers as required until the request is complete.

---



|                       |   |
|-----------------------|---|
| <b>PSBname</b>        | <p>This field is used only if the THREADUSAGE gateway startup parameter is set to TABLE or TRANSACTION. Refer to <a href="#">Supplying Gateway Startup Parameters on page 57</a> for more information.</p> <p>The PSBNAME parameter can be overridden at runtime. Refer to <a href="#">Chapter 2, Operating the Service Gateway for IMS/DB, on page 49</a> for more information.</p> <p>Specify the name of the PSB to be scheduled by the TIBCO Service Gateway for IMS/DB when this is the first IMS table accessed in a transaction.</p> <p>If a PSBname is not specified in the first IMS table accessed in a transaction, the Gateway schedules the PSB specified by the PSBNAME gateway startup parameter.</p> <p>If the PSBNAME gateway startup parameter is not specified, the default S6BIMSS is used. This PSB is then scheduled for the duration of the TIBCO Object Service Broker transaction.</p> <p>Depending on the setting of the THREADUSAGE parameter, the following occurs:</p> |
|                       |   |
| TRANSACTION           | A PSB specified by a table in a subsequent TIBCO Object Service Broker IMS access request is ignored.   |
| TABLE                 | All PSBs specified by tables in subsequent TIBCO Object Service Broker IMS access requests are checked. The PSB must be the same as the one specified in the first IMS table accessed or it must be blank for the TIBCO Object Service Broker transaction to proceed.   |
|                       |   |
| <b>Browse PSBname</b> | <p>This field is used only if the Gateway is running in either BMP, DRA, or CICS mode with the THREADUSAGE startup gateway parameter set to TRANSACTION. Refer to <a href="#">Supplying Gateway Startup Parameters on page 57</a> for more information.</p> <p>Specify the name of the Browse PSB if you want to use a read-only PSB (for example, a PSB with a PROCOPT=GP) when running an TIBCO Object Service Broker transaction in browse mode.</p>   |

## Task E: Define a Location Parameter

---

You use this section of the table definition screen to define a location parameter for an IMS table. You use a location parameter to access IMS or DB2 data through a peer Gateway associated with another Data Object Broker (remote node). If you do not need to access remote data, use the **D** line command to delete the parameter. If you always access the IMS or DB2 table remotely, the node from which you request the access can have either a minimal or a full definition.

### Minimal and Full Node Definitions

Define data parameters on the full definition, not a minimal definition.

#### Minimal Definitions

A minimal definition with a location parameter means you always access data at a remote node. A minimal definition consists of the following:

- The table name, which must be the same at both locations.
- The location parameter, which must be the same at both locations. The name of the remote node where the full definition is located *must* be supplied in the **Default** field, **Src** field, or **Src** and **Sourcename** field.

The table type specified in a minimal definition does not have to match the table type of the full definition on the remote node.

#### Full Definitions

A full table definition with a location parameter means you can access data at either the local or the remote node.

The table type of the full definition must match the data on the local node. For example, a full definition of type TDS used to access TDS data on the local node can also be used to access an IMS table with the same name on a remote node.



If the Execution Environment is non-z/OS, the non-z/OS Data Object Broker requires a minimum or a full TDS table definition. IMS tables cannot be defined on a non-z/OS Data Object Broker.

See Also

*TIBCO Object Service Broker Managing Data* for more information on location parameters and minimal table definitions.

# Task F: Select IMS Segments

---

After you select an IMS database, you must select segments from it. To do this, you must determine the access path required to retrieve the data you need. Refer to [Effects of Segment Selection on page 123](#) for more information.

The root segment is automatically included in the access path and already appears in the **Segment Name** field on the IMS Table Definition screen with a level of 0. Additional segments that you select must be along one access path (that is, you can choose only one segment from each level, as long as you also choose its parent).

## Selecting Additional Segments

To select additional segments, complete the following steps:

1. Position the cursor beside the root segment name and press PF6 to display existing child segments.

A screen similar to the one shown below appears.

---

Use 'S' to select a child segment:

| Level | Segment Name |
|-------|--------------|
| --    | -----        |
| 1     | ADDRESS      |

PFKEYS: 3=END 12=END ENTER=SELECT

---

2. Select one segment from the list of child segments by typing an **s** beside it and pressing Enter.

A screen similar to the one shown below appears with the selected child segment listed under the parent segment.

```
COMMAND==>                                TABLE DEFINITION

      Table: CUST_ORDERS      Type: IMS    Unit: USR40
      DB Name:
      Server ID: DEFAULT      Server Type: IMS  Optimize Update:
      PSBname: OSTRXDRP      Browse PSBname: OSTRXDRQ

      Location Parm      Typ Syn Len Dec Default      Src Sourcename
      -----
      _ LOCATION              I   C  16  0

      Level      Segment Name      Fields Chosen      Order From Segment
      --      -----
      _ 0      CUSTROOT
      _ 1      ADDRESS

      Event Rule      Type Access      Parms Chosen      Order From Segment
      -----
      _
      _
      _
PFKEYS: 3=SAVE 12=CANCEL 22=DELETE 13=PRINT 6=CHILD SEGMENTS 2=DOC 4=DB NAMES
```

- 3. Repeat this process until you select all the segments that you require along the desired access path.

## Task G: Select IMS Fields

After selecting an IMS database and the required segments, you must select and define all the IMS fields for each selected segment. Before you select fields, you must understand the effects of field selection on IMS processing. Refer to [Effects of Field Selection on page 124](#) for more information.

To select fields from a segment shown on the IMS Table Definition screen, shown in [Task B: Invoke the Table Definer on page 97](#), type an **S** beside the segment and press Enter. A screen similar to the following appears.

### IMS Field Selection Screen

|                    |        |                |                |       |                        |     |     |  |
|--------------------|--------|----------------|----------------|-------|------------------------|-----|-----|--|
| Table: CUST_ORDERS |        | Type: IMS      | Unit: USR40    |       |                        |     |     |  |
| DB Name: CUSTOMER  |        |                |                |       |                        |     |     |  |
| Segment: CUSTROOT  |        | Min length: 41 | Max length: 41 |       |                        |     |     |  |
| IMS Name           |        | -----          | IMS            | ----- | --- Metadata Field --- |     |     |  |
|                    | Offset |                | Syn            | XSyn  | Syn                    | Dec | Rqd |  |
| Metadata Name      | Idx    | Len            | Val            |       | Typ                    | Len | Ord |  |
| -----              | ---    | ----           | -              | -     | ----                   | -   | -   |  |
| — CUST#            | N      | 0              | 5              | C     | C                      |     |     |  |
| — CUSTNAME         | N      | 5              | 25             | C     | C                      |     |     |  |
| — ODATE            | N      | 30             | 10             | C     | C                      |     |     |  |
| — XPCODE           | Y      | 0              | 9              | C     | C                      |     |     |  |

(S=Select K=Key P=Parameter V=Access Value D=Unselect I=Insert)  
 PFKEYS: 3=END 12=CANCEL 6=VIEW SUB/BASE IX

The IMS Field Selection screen is divided into the following sections:

- Header
- IMS
- Metadata Field

If you insert a new field (using the **I** line command), you must complete the IMS and the Metadata Field sections. If you select the field, you can modify the Metadata Field section if necessary.

## Interpreting the Header Section

The following fields cannot be modified; they appear for information only:

|                   |   |
|-------------------|---|
| <b>Table</b>      | Displays the table name specified when you invoked the Table Definer.   |
| <b>Type</b>       | Displays the table type IMS.  |
| <b>Unit</b>       | Displays the user unit associated with the table.   |
| <b>DB Name</b>    | The name of the IMS database upon which you are basing your IMS table.  |
| <b>Segment</b>    | The name of the IMS segment for which you are selecting or defining fields.   |
| <b>Min Length</b> | <p>For variable length segments, this is the minimum length of the segment in bytes. For fixed length segments, it is the actual length of the segment in bytes.</p> <p><b>Note</b> If Optimize Update = Y, this is the minimum length that must be defined to TIBCO Object Service Broker for each segment starting at the segment where the primary key is defined.</p> |
| <b>Max Length</b> | For variable length segments, this is the maximum length of the segment in bytes. For fixed length segments, it is the actual length of the segment in bytes.   |

## IMS Field Section

The following fields are protected unless you are creating a field layout definition, in which case you can modify them.

|                      |   |
|----------------------|---|
| <b>IMS Name</b>      | Displays the name of the IMS field that is part of the segment that you are defining.   |
| <b>Metadata Name</b> | <p>Displays the TIBCO Object Service Broker name for the IMS field that is part of the segment that you are defining. The default is the name in the <b>IMS Name</b> field.</p> <p>Valid entries include character strings of up to 16 characters, beginning with a letter (A-Z) or a special character (\$ or #), and continuing with more letters, special characters, digits (0-9), or underscore characters(_).</p> |
| <b>Idx</b>           | Y specifies that this field is an IMS Secondary Index field. N means it is not. You can specify only a secondary index field to be a key, access value, or parameter, but not a field. You can specify up to five fields as a composite secondary index. Refer to <a href="#">Defining Secondary Indexes on page 119</a> for more information.  |
| <b>Offset</b>        | The offset of the IMS field from the beginning of the segment starting at offset 0. The default is 0. You cannot modify the offset for fields defined in the DBD. You must supply the length when inserting a new field, and it must be within the boundary of the segment.   |
| <b>Len</b>           | The length of the IMS field in bytes. The default is 0. You cannot modify the length for fields defined in the DBD. You must supply the length when inserting a new field, and it must be within the boundary of the segment.   |
| <b>Syn</b>           | <p>Displays the syntax of the IMS field as defined in the DBD. You cannot modify this field, it is for information only. Valid entries:</p> <ul style="list-style-type: none"> <li>• B (Binary)</li> <li>• C (Character)</li> <li>• F (Full word binary)</li> <li>• H (Half word binary)</li> <li>• P (Packed decimal)</li> </ul>   |

|             |   |
|-------------|---|
| <b>Val</b>  | Specifies whether there is an access value associated with the field. You use an access value to access a subset of the segment occurrences based on the value of an IMS field. Refer to <a href="#">Defining Access Values on page 119</a> for more information. Valid entries are: <ul style="list-style-type: none"><li>• Y—An access value is specified for the field</li><li>• Null—No access value is specified for the field</li></ul> |
| <b>Xsyn</b> | Identifies the external syntax of the field. This is the syntax the Gateway expects when retrieving IMS data and that the field is converted to when data is sent back to IMS. Refer to the table under <a href="#">Mapping Data Types on page 121</a> for valid values.  |

Metadata Field Section

The following fields are found in the Metadata Field section of the screen:

|            |   |
|------------|---|
| <b>Typ</b> | Specifies the TIBCO Object Service Broker semantic type of the field. You can use any valid semantic type and syntax combination. Valid values are defined in the <i>TIBCO Object Service Broker Programming in Rules</i> manual. Refer to <a href="#">Mapping Data Types on page 121</a> for more information. |
| <b>Syn</b> | Specifies the TIBCO Object Service Broker syntax of the field. You can use any valid semantic type and syntax combination. Valid values are defined in the <i>TIBCO Object Service Broker Programming in Rules</i> manual. Refer to <a href="#">Mapping Data Types on page 121</a> for more information.        |
| <b>Len</b> | Specifies the TIBCO Object Service Broker length of the IMS field in bytes. Refer to <a href="#">Mapping Data Types on page 121</a> for more information.   |
| <b>Dec</b> | The number of digits to the right of the decimal point. Valid only for syntax P. The number of decimal places must be smaller than twice the length of the field. For syntax B, C, or V, or semantic type C, the value must be 0.   |



---

|            |   |
|------------|---|
| <b>Ord</b> | <p>Specifies the order in which occurrences are returned to TIBCO Object Service Broker. If you do not specify a value, occurrences are returned in the order they are retrieved from IMS. Valid entries:</p> <ul style="list-style-type: none"><li>• A (ascending)</li><li>• D (descending)</li><li>• blank (no ordering)</li></ul> <p>If you specify ordering in this field, the following occurs:</p> <ul style="list-style-type: none"><li>• For a qualified GET or FORALL, all segment occurrences matching the selection criteria are retrieved and sorted by the Execution Environment.</li><li>• For an unqualified GET or FORALL, all segment occurrences are retrieved and sorted by the Execution Environment before data is passed to the rule or tool.</li></ul> |
| <b>Rqd</b> | <p>Specifies if you must enter data in this field in the IMS table definition.</p> <p>Valid Entries:</p> <ul style="list-style-type: none"><li>• Y (required)</li><li>• N (not required)</li><li>• blank (not required)</li></ul>   |

---

After selecting and defining all necessary fields, press PF3 to return to the IMS Table Definition screen (shown below).

Sample Defined Table Definition Screen

COMMAND==>TABLE DEFINITION

Table: CUST\_ORDERSType: IMSUnit: USR40

DB Name:

Server ID: DEFAULTServer Type: IMSOptimize Update:

PSBname: OSTRXDRPBrowse PSBname: OSTRXRQ

| Location | Parm | Typ | Syn | Len | Dec | Default | Src | Sourcename |
|----------|------|-----|-----|-----|-----|---------|-----|------------|
| LOCATION |      | I   | C   | 16  | 0   |         |     |            |

| Level | Segment Name | Fields Chosen | Order | From Segment |
|-------|--------------|---------------|-------|--------------|
| 0     | CUSTROOT     | PRODUCT#      | 1     | PRODORD      |
| 1     | ORDER        | QUANTITY      | 2     | PRODORD      |
| 2     | PRODORD      |               |       |              |

| Event Rule | Type | Access | Parms Chosen | Order | From Segment |
|------------|------|--------|--------------|-------|--------------|
|            |      |        | CUSTNO       | 1     | CUSTROOT     |
|            |      |        | ORDER#       | 2     | ORDER        |

PFKEYS: 3=SAVE 12=CANCEL 22=DELETE 13=PRINT 6=CHILD SEGMENTS 2=DOC 4=DB NAMES

On the IMS Field Selection screen, shown in [Task G: Select IMS Fields on page 109](#), place the cursor beside the IMS field you want to select and use one of the following line commands:

- S

Selects a field to be a field in the IMS table.

The number of fields you can define depends upon the length of the sum of all fields, primary keys, and data parameters and control information. This sum must be less than or equal to 3915 bytes. For a detailed explanation of the formula used to calculate the total number of bytes of all fields, refer to the *TIBCO Object Service Broker Managing Data* manual.

The number of fields you can access depends upon the CTABLESIZE Data Object Broker parameter. You use the ESTIMATETBLDFN, rule described in [Estimating the CTABLESIZE Parameter on page 72](#), to estimate the size of this parameter.
- K

Selects a field to be a primary key field. You can select up to four IMS fields as composite primary keys.

- 
- P** Selects a field to be a parameter. You can choose up to four IMS fields as parameters, usually from four different segment levels. The fields you select appear on the initial IMS Table Definition screen in the Parms Chosen segment of the screen.<sup>a</sup>
- 
- V** Associates an access value to the field. Refer to [Defining Access Values on page 119](#) for more information.<sup>a</sup>
- 
- D** Removes a field from the IMS table definition. For a new field that added using the **I** line command. This line command deletes the field.
- 
- I** Inserts a field into the IMS table definition after the field where the cursor is positioned. You must provide the values for the IMS field and MetaStor field sections of the definition. These fields cannot overlap in a segment; if they do, no updates are permitted.

The S6BIMSU extract program extracts the characteristics of all fields defined in the DBD as described in [Task A, Extract IMS database information, on page 50](#). Use the **Max Length**, **IMS Off**, and **IMS Len** fields to determine if all fields are defined in the DBD. If fields are missing from the definition, consider inserting these fields (referring to the appropriate copy library member for the full definition); refer to [Replace \(Update\) Processing on page 134](#) and [Insert Processing on page 135](#) for more information.

---

a. To use an IMS secondary index to retrieve IMS data, you must select the index field (identified with a Y in the Idx field) as a parameter or access value. The database PCB defined in the Gateway PSB must also contain a PROCSEQ or INDICES parameter. Refer to [Defining Gateway Program Specification Blocks on page 26](#) for more information.

If necessary, you can modify the display order of all selected fields and parameters, except for the primary key field (denoted by a 1 in the **Order** field of the Fields Chosen area of the screen). To change the order, type new numbers into the Order field in either the Fields Chosen area and the Parms Chosen area and press Enter.

After selecting the necessary IMS fields, press Enter to display the default name, type, syntax, and length of these fields in the TIBCO Object Service Broker Field segment. A screen similar to the one shown below appears.

IMS Field Selection Screen

|                    |        |                 |                            |     |
|--------------------|--------|-----------------|----------------------------|-----|
| Table: CUST_ORDERS |        | Type: IMS       | Unit: USR40                |     |
| DB Name: CUSTOMER  |        |                 |                            |     |
| Segment: CUSTROOT  |        | Min length: 41  | Max length: 41             |     |
|                    |        |                 |                            |     |
| IMS Name           |        | ----- IMS ----- | ----- Metadata Field ----- |     |
|                    | Offset | Syn             | XSyn                       | Syn |
| Metadata Name      | Idx    | Len             | Val                        | Typ |
|                    |        |                 |                            | Len |
|                    |        |                 |                            | Ord |
|                    |        |                 |                            |     |
| — CUST#            | CUST#  | N               | 0                          | 5   |
| — CUSTNAME         |        | N               | 5                          | 25  |
| — ODATE            |        | N               | 30                         | 10  |
| — XPCODE           |        | Y               | 0                          | 9   |

(S=Select K=Key P=Parameter V=Access Value D=Unselect I=Insert)  
PFKEYS: 3=END 12=CANCEL 6=VIEW SUB/BASE IX



Default TIBCO Object Service Broker values are supplied only for fields that are defined in the DBD of the database.

See Also *TIBCO Object Service Broker Parameters* about the CTABLESIZE Data Object Broker parameter.

## Task H: Define Event Rule Information

You use event rules to validate data and automatically trigger other events based on specific update and retrieval access to IMS tables. Event rules are always called when the table is accessed in the type of access specified. All rules applying to a specific access are executed in the order they appear in the scrollable Event Rule segment of the IMS Table Definition screen.

### Defining Fields

You define the fields in the Event Rule segment as follows:

|   |   |
|---|---|
| Type  | Specifies the type of the event rule:   |
| <div> <div>V</div> <div> <div>Validation Rule</div> <div>Database updates are prohibited during the validation process. The rule must be a function that returns Y (yes) if the validation is successful, N (no) if the validation is not successful, or a message explaining why it is not successful.</div> </div> </div> |   |
| <div> <div>T</div> <div> <div>Trigger Rule</div> <div>There are no restrictions on coding, other than the rule must not be a function, it cannot change the contents of the triggering row, and it cannot use the TRANSFERCALL statement. Nested triggers are possible.</div> </div> </div>                                 |   |
| Access  | The type of access, or manipulation, to be performed on the data, causing the event to be executed. |

| Validation Rules                        | Trigger Rules                          |
|---|--|
| W - Any Write (Insert, Replace, Delete) | W- Any Write (Insert, Replace, Delete) |
| I - Only Insert                         | I - Only Insert                        |
| R - Only Replace                        | R - Only Replace                       |

| Validation Rules | Trigger Rules     |
|------------------|-------------------|
| D - Only Delete  | D - Only Delete   |
|                  | G - Any Retrieval |

See Also *TIBCO Object Service Broker Managing Data* for more information on event rules.

## Defining Secondary Indexes

---

You can define up to five fields of any syntax as secondary index fields for an IMS table definition. Consider the following when defining secondary indexes on IMS fields:

- Index fields can be selected only as **S**, **K**, **V**, or **P** fields; however, only base index fields can have access values (V).
- Support for mixed case character secondary index fields is available.
- If you toggle from **Base Index** field to sub index fields (pressing PF6), all sub index fields are automatically selected. It is a Gateway requirement that all sub index fields are included.
- Offset for index (base or sub index fields) always start at 0. Offset for subfields are calculated relative to the base index.

### Defining Access Values

You can access a subset of segment occurrences by supplying a value for one or more IMS fields. This restricts which segment occurrences are returned to TIBCO Object Service Broker. The value (specified in upper or lowercase) is used to search the database with an equality condition. For example, you could choose to return all occurrences where CUSTNAME = Smith.

To define an access value, use the following procedure.

1. On the IMS Field Selection screen shown in the [Task G: Select IMS Fields on page 109](#), type **v** beside the IMS field or fields for which you want to specify an access value and press Enter.
2. On the Access Value screen, type a valid IMS field value at the cursor position and press PF3 to save or PF12 to cancel.

If you selected more than one field, the Access Value screen is refreshed when you press PF3 or PF12, and you can specify the value for the next selected field.

3. To remove an access value, blank out the value in the Access Value screen or type **D** beside the field you are modifying and press Enter.
4. Fields with access values are not automatically selected as fields of the IMS table. To select an access value field as a field of the table, type an **S** beside it and press Enter.

Fields containing access values are identified with a Y in the **Val** field on the IMS Field Selection screen.

### Sample Access Value Screen

Enter the access value for CUSTNAME

PFKEYS: 3=END 12=CANCEL

### Defining Multiple Record Formats

You define segments with more than one record format to TIBCO Object Service Broker by supplying access values. When an IMS segment contains a different number of fields depending on the value of another field within the segment, you must create one IMS table definition for each access value you require.

For example, the table below illustrates DEFN1 and DEFN2 using different access values to define RECTYPE = 1, where the segment contains three numeric fields, and RECTYPE = 2, where the segment contains two character fields.

#### Defining Multiple Record Formats to TIBCO Object Service Broker

| IMS Table | Access Value for Field RECTYPE | IMS Table Fields |   |        |
|-----------|--------------------------------|------------------|---|--------|
| DEFN1     | 1                              | Fld1             | 5 | Packed |
|           |                                | Fld2             | 5 | Packed |
|           |                                | Fld3             | 5 | Packed |



| IMS Table | Access Value for Field RECTYPE | IMS Table Fields |    |           |
|-----------|--------------------------------|------------------|----|-----------|
| DEFN2     | 2                              | Fld1             | 10 | Character |
|           |                                | Fld2             | 5  | Character |

## Mapping Data Types

The table below displays default mapping of IMS data types from the DBD to TIBCO Object Service Broker semantic types and syntax. Valid TIBCO Object Service Broker semantic types and syntax are described in *TIBCO Object Service Broker Programming in Rules*.

### IMS to TIBCO Object Service Broker Default Data Mapping

| IMS DBD Syntax (SynField) |                          | TIBCO Object Service Broker Semantic Type | TIBCO Object Service Broker Syntax | TIBCO Object Service Broker Length |
|---------------------------|--------------------------|---|------------------------------------|------------------------------------|
| B                         | Binary                   | Q   | P                                  | 3 (for Binary 2) 5 (for Binary 4)  |
| C                         | Character                | S   | C or V                             | maximum of 254                     |
| F                         | Fullword binary, 4 bytes | Q   | P                                  | 5                                  |
| H                         | Halfword binary, 2 bytes | Q   | BP                                 | 3                                  |
| P                         | Packed decimal           | Q   | P                                  | maximum of 8                       |

IMS Data Translation

This table displays the valid TIBCO Object Service Broker syntax and length to use when defining new fields or modifying the default syntax and length of existing fields. Valid TIBCO Object Service Broker semantic types are described in *TIBCO Object Service Broker Programming in Rules*.

| IMS External Syntax (Xsyn) |   | IMS Length           | TIBCO Object Service Broker Syntax | TIBCO Object Service Broker Length |
|----------------------------|---|----------------------|------------------------------------|------------------------------------|
| A                          | Alphabetic (uppercase only)   | 1 - 254              | C or V                             | 1 - 254                            |
| B                          | Binary, signed  | 1 - 4                | P                                  | 3, 5 <sup>a</sup>                  |
| C                          | Character   | 1 - 254              | C or V                             | 1 - 254                            |
| G                          | Packed, neutral - (X'0F') sign when positive (X'0D') sign when negative | 1 - 8                | P                                  | 1 - 8                              |
| H                          | Hexadecimal   | 1 - 147 <sup>b</sup> | RD                                 | 5 - 151                            |
| K                          | Binary, unsigned  | 1 - 4                | P                                  | 3, 5 <sup>a</sup>                  |
| M                          | Numeric, unsigned   | 1 - 8                | P                                  | 1 - 8                              |
| N                          | Numeric, zoned  | 1 - 8                | P                                  | 1 - 8                              |
| O                          | Packed, no sign   | 1 - 8                | P                                  | 1 - 8                              |
| P                          | Packed decimal  | 1 - 8                | P                                  | 1 - 8                              |
| U                          | Packed, unsigned  | 1 - 8                | P                                  | 1 - 8                              |
| V                          | Variable length character (mixed case)                                  | 1 - 254              | V                                  | 1 - 254                            |
| Z                          | X'00' filled character  | 1 - 254 <sup>c</sup> | V                                  | 1 - 254                            |

- a. You must specify the appropriate number of decimal places, if necessary.
- b. The hexadecimal value of data appears. For example, a two-byte field (X'C1C2') appears in a four-byte variable-length TIBCO Object Service Broker field (C1C2).
- c. The remainder of a variable length character field is padded with binary zeros, instead of blanks.

**Note:** The date (D) semantic type is not valid for IMS tables.

## Mapping Considerations

---

To choose an optimal access path, it is critical that the person defining IMS table definitions be familiar with the structure of the IMS database.

### Effects of Segment Selection

The following factors must be considered when selecting IMS segments:

#### Table Definitions with More than One Segment Defined

When more than one segment is defined in an IMS table definition, the Gateway returns data only when all segments exist in the specified hierarchy. For example, if the CUSTROOT and ORDER segments are defined, only information on customers that have orders is retrieved.

#### Retrieving and Updating Specific IMS Data

To directly retrieve and update specific IMS data, you should have one IMS table definition for each segment that can be uniquely identified. To do this, complete the following:

- Parameterize parent segments. These segments are used for positioning only.
- Select the unique sequence key of the segment you are defining as the TIBCO Object Service Broker primary key. This segment should be the last segment in the access path.

[Appendix B, Samples of Recommended Table Definitions, on page 145](#) explains how to define an IMS table definition for an IMS database containing unique segments.

### Retrieving and Updating Non-unique IMS Data

To directly retrieve and update non-unique IMS data, complete the following:

1. Parameterize unique parent segments. These segments are used for positioning only.
2. Define the TIBCO Object Service Broker primary key on the first non-unique segment in the access path.

[Appendix B, Samples of Recommended Table Definitions, on page 145](#) explains how to define an IMS table definition for an IMS database containing unique segments.



To modify non-unique IMS data, specify Optimize Update = Y in the IMS Table Definition screen.

For definitions with non-unique IMS segments, the Gateway requires at least two identical PCBs defined for the database in question. Refer to [Retrieval Processing on page 132](#) for more information.

### Parameter Values for Parameterized TIBCO Object Service Broker Tables

You must provide parameter values for parameterized tables. If these parameters map to unique IMS sequence keys, the Gateway can use this information to build a qualified Segment Search Argument (SSA) so IMS can directly retrieve the requested data.

### Retrieving Segment Occurrences

If you define one IMS table definition for each unique segment, all segment occurrences can easily be retrieved using a combination of the tables. The root segment IMS table could be used as a TIBCO Object Service Broker PRM table in a nested FORALL loop to retrieve all child segments. All segments at each level can be retrieved using its parent segment IMS table definition.

When you define one IMS table definition for each unique segment, and if this data is moved into TIBCO Object Service Broker TDS tables, you can map each IMS table definition to a TDS table definition.

### Hierarchical Paths and Required Segments

To sweep large amounts of IMS data, it could be more efficient to determine the necessary hierarchical paths and build an IMS table containing all required segments, with the TIBCO Object Service Broker primary key defined on the first segment.

## Effects of Field Selection

The following factors must be considered when selecting IMS fields for an IMS table definition:

## Unique Sequence Keys

TIBCO Object Service Broker expects the combination of parameters and primary key fields to uniquely identify an occurrence. When possible, select unique sequence keys as parameters and primary keys. To determine the unique sequence key of each segment, browse the @IMSSEGS(*dbname*) table.

## Definitions in the IMS DBD

A field selected as a TIBCO Object Service Broker primary key, parameter, or access value must be defined in the IMS DBD. The Gateway uses these fields to build an IMS access request in conjunction with your selection criteria.

## Primary Keys

Each IMS table definition must have a primary key. Inserts, replaces, and deletes occur at the segment level at which the primary key is defined.

## Sequence Keys

Identify unique sequence keys to TIBCO Object Service Broker by defining them as parameters, access values, or primary keys. If you select a unique sequence key as a field of an IMS table, it can cause unexpected results if this field is modified. Refer to [Using the Table Browser and Table Editor on page 128](#) for details.

## Performing INSERT, DELETE, and REPLACE

To perform INSERTs, DELETEs, REPLACEs, or if Optimize Update=Y, *all* fields for the segment level at which the primary key is defined must be fully defined to TIBCO Object Service Broker, including fields that are not defined in the DBD.



If your definition contains more than one segment type under the TIBCO Object Service Broker primary key, all fields that belong to each segment type must be fully defined to TIBCO Object Service Broker.

## Performing Replaces

To perform REPLACEs, fields defined in the IMS table starting at the level at which the TIBCO Object Service Broker primary key is defined must not overlap. Use the **Off** and **Len** fields on the IMS Field Selection screen to determine if fields overlap.

**Large Fields and Transaction Length**

Including large IMS fields, close to the maximum size of 3915 bytes, in your table definition increases transaction length due to an increased number of commits. This can affect the performance of your application.

## Chapter 4      **Processing IMS Data**

This chapter shows you how to access the TIBCO Object Service Broker IMS tables.

### Topics

---

- [Accessing TIBCO Object Service Broker IMS Tables, page 128](#)
- [Using Rules, page 131](#)
- [Understanding Gateway Behavior, page 136](#)

## Accessing TIBCO Object Service Broker IMS Tables

---

When accessing IMS data from TIBCO Object Service Broker, TIBCO Object Service Broker requests are translated into Segment Search Arguments (SSAs), which are used in a DL/I call to IMS. SSAs are built based on parameterized fields, access values, and the selection criteria specified for primary keys and any field defined in the DBD. The DBD used when data is accessed must be identical to the DBD used when the TIBCO Object Service Broker IMS table is defined.

IMS field data types are translated to the TIBCO Object Service Broker field types defined in the IMS table.

You can access the data using:

- The Table Browser and Table Editor
- The rules language

### Using the Table Browser and Table Editor

You can browse and edit an IMS table in the same way you would browse or edit any other TIBCO Object Service Broker table with the following exceptions:

#### Optimize Update=Y

If Optimize Update=Y, you cannot update or order the segment occurrences of the table.

#### Modifying Primary Key Values

If you modify the value of the primary key of a table in the Table Editor, a DELETE followed by an INSERT is issued on your behalf. If the segment the primary key is defined at is a parent, all child segments associated with that parent are deleted. Only the child segment occurrence displayed in the Table Editor is inserted back into the database.

#### Field of Syntax C or V Longer than 260 Bytes

If your table definition contains fields of syntax C or V that are longer than 260 bytes:

- You must use the Single Occurrence Editor (SOE) from the Table Editor to edit them.
- You must use **SELECT LIKE** instead of **SELECT** to access fields of this length.



## Using the Single Occurrence Editor

Using the Single Occurrence Editor from the Table Browser begins a dependent transaction in TIBCO Object Service Broker, assigning a new Gateway. Therefore, to use the Single Occurrence Editor from the Table Browser, you must have one of the following running with the same SERVERID:

- Gateway accessing IMS data only
- Gateway accessing both IMS and DB2 data
- Gateway under CICS DL/I

However, you can use the Single Occurrence Editor from the Table Editor without this condition, since this does not begin a dependent TIBCO Object Service Broker transaction.



The Table Browser does *not* hold locks on TDS data; however, this is not true for IMS tables. Locking of IMS data is determined by the PROCOPT parameter for the database PCB in the Gateway PSB. Refer to [Defining Gateway Program Specification Blocks on page 26](#).

Using the Single Occurrence Editor from the Table Browser can cause undetectable parent-child deadlocks.

## Unique Sequence Keys of Child Segments

If you include the unique sequence key of a child segment as a field of an IMS table (instead of a parameter, access value, or primary key) and you modify the value of this field the following occurs:

- The Gateway attempts to retrieve the segment occurrence with a Get Hold Unique (GHU) using the unique sequence key of both the parent and the modified child segment.
- If the newly specified combination of unique sequence keys exists, the entire segment occurrence is replaced with the data displayed in the Table Editor or the Single Occurrence Editor. Be careful that the replaced segment occurrence is the occurrence you intended to update.
- If the newly specified combination of unique sequence keys does not exist, an error message is issued stating that the update failed because the occurrence could not be found.

You can be sure only that the proper replacement occurs if unique sequence keys of child segments are identified as key fields to TIBCO Object Service Broker.

## Browsing IMS Tables

You can use the Table Browser to browse a defined IMS table by typing a table name next to the BR browse table option, for example:

CUST\_ORDERS<Enter>

The sample screen below shows IMS data presented in TIBCO Object Service Broker table format. The IMS table is named CUST\_ORDERS and the parameter values for CUSTNO and ORDER# are both 000001.

### Browsing an IMS Table

---

```
BROWSING TABLE      :  CUST_ORDERS(000001,000001)
COMMAND ==>

                                SCROLL:  P

PRODUCT#  QUANTITY
-----  -
- 000001      99
- 000002     999
- 000003      10
- 000004      10

PFKEYS: 1=HELP 5=FIND NEXT 9=RECALL 18=EXCLUDE 13=PRINT 3=END 14=EXPAND
At TOP
```

---

See Also     *TIBCO Object Service Broker Managing Data* for more information on browsing tables.

## Using Rules

---

Accessing IMS data using the TIBCO Object Service Broker rules language is similar to accessing TIBCO Object Service Broker data. The main difference is in the way IMS uses your selection criteria in conjunction with the IMS table definition to interpret the request.

The following sections outline any differences encountered while using rules, point out normal rules behavior that must be considered when building applications, and explain how your rules selection criteria translates into DL/I calls.

### Transaction Processing

If you issue a rule language EXECUTE statement within a main (parent) transaction, it creates another transaction stream (child transaction), to a maximum of ten streams. The number of streams allowed in a TIBCO Object Service Broker transaction depends on the TRANMAXNUM Execution Environment parameter, which has a default of nine streams. Each transaction stream in TIBCO Object Service Broker accessing IMS data requires its own TIBCO Service Gateway for IMS/DB or TIBCO Service Gateway for IMS/DB under CICS DL/I task.



Ensure your system administrator is aware of the number of TIBCO Service Gateway for IMS/DB or TIBCO Service Gateway for IMS/DB under CICS DL/I tasks required to accommodate all transaction streams accessing IMS data in a single transaction.

### POOLSIZE and CTABLESIZE Parameters

The number of IMS tables you can access per transaction depends on the POOLSIZE gateway parameter, and the CTABLESIZE Data Object Broker parameter. If the default parameter sizes are used, you can access at least 16 IMS tables per transaction; more, depending on the size of the IMS table definitions. Refer to [Estimating the CTABLESIZE Parameter on page 72](#) for more information. Refer to *TIBCO Object Service Broker Parameters* for more information about parameters.

### TRANSFERCALL and Data Object Broker DISPLAY & TRANSFERCALL Statements

Using TRANSFERCALL or DISPLAY & TRANSFERCALL statements in a rule minimizes the number of Gateways and reduces the possibility of IMS locking contention.

## THREADUSAGE Parameters

If your TIBCO Service Gateway for IMS/DB is executing in a CICS Execution Environment, when the THREADUSAGE gateway startup parameter is set to TRANSACTION or TABLE, the PSB specified by the first IMS table accessed in a TIBCO Object Service Broker transaction is scheduled for the duration of that transaction.

If your TIBCO Object Service Broker transaction requests access to IMS data using more than one IMS table definition, the following occurs:

- If THREADUSAGE=TRANSACTION, the PSB scheduled at the start of a transaction is assumed to be valid for all subsequent access requests. If a subsequent IMS table specifies a different PSB, it is ignored.
- If THREADUSAGE=TABLE, the PSB scheduled at the start of a transaction is still valid for all subsequent access requests. However, the PSB specified by subsequent IMS table definitions must be the same as the scheduled PSB or the subsequent definitions must not specify a PSB. If a different PSB is encountered, the transaction fails.

## Retrieval Processing

The Gateway uses PATH calls to access IMS data. Only one PATH call is required for each GET or FORALL in your rule that is retrieving a sequence of segments in a hierarchical path. An occurrence is returned only if data is available for all segments defined in the path.

The Gateway includes all parameter values and access values in the SSA with an equality qualification. Additionally, the Gateway uses the selection criteria specified for primary keys and for IMS fields that are defined in the DBD, providing the selection criteria meets all the following conditions:

- Selection criteria is specified with an AND (&) condition.
- One of the following operators: =, >, >=, <, or <= is specified for a primary key and an IMS field defined in the DBD.
- For HDAM databases only, an equality operator is specified for a primary key. HDAM databases use a random storage method, therefore, if another operator is specified, the Gateway does the following:
  - For GET processing, it reads all segment occurrences in the access path, until it finds the first occurrence satisfying the selection criteria.

- For FORALL processing, it reads all segment occurrences in the access path to find all occurrences satisfying the selection criteria.



Use of the the disjunctive logical operator "OR" ("|") or the "NOT" ("¬") logical operator will result in a sweep of the IMS data without field qualification via an IMS Segment Search Argument (SSA).

When a TIBCO Object Service Broker transaction runs in browse mode, locks are not taken on the TIBCO Object Service Broker data. Locking of IMS data is determined by the PROCOPT parameter for the database PCB in the Gateway PSB. Refer to [Defining Gateway Program Specification Blocks on page 26](#) for more information.

## GET Statement

A GET statement retrieves the first occurrence in the IMS table satisfying the specified selection criteria.

A GET...ORDERED statement retrieves all IMS data satisfying the selection criteria, and sorts it in the Execution Environment before returning the first occurrence.

## FORALL Statement

A FORALL statement is a looping construct that processes a set of occurrences. The body of the loop consists of the statements to be executed for each occurrence satisfying the selection criteria. FORALL statements can be nested provided they refer to different table names.

A FORALL returns occurrences to TIBCO Object Service Broker in the order in which IMS passes them. To get the data in a different order, include an ORDERED clause in the FORALL. TIBCO Object Service Broker orders only occurrences specified in the selection criteria.

## Retrieving Non-unique Segments

IMS tables containing non-unique segments (no key, or a multiple sequence key) require their own database PCB. This enables the Gateway to maintain its position in the IMS database when accessing non-unique IMS data.



Ensure that your IMS database administrator is aware of the maximum number of GET and FORALL statements that access non-unique IMS tables in a single TIBCO Object Service Broker transaction. The database administrator needs to know to which IMS database these tables map, to determine the number of database PCBs required for each database defined in the Gateway PSB. Refer to [Chapter 2, Operating the Service Gateway for IMS/DB, on page 49](#) for more information.

## Replace (Update) Processing

A REPLACE request against an IMS table modifies the last TIBCO Object Service Broker IMS occurrence retrieved. All segments defined in the access path are replaced, starting with the segment where the primary key is defined. Parameterized segments are used for positioning only.

## Considerations

You can modify all fields selected from each segment defined in the access path, starting with the segment where the primary key is defined. Consider the following when modifying fields:

- You cannot modify unique sequence key fields. IMS requires you to delete the segment occurrence and re-insert it into the database with the new key value. The consequences of modifying the value of a child segment's sequence key (that was selected as a field of the IMS table definition) are described in the fourth point in [Using the Table Browser and Table Editor on page 128](#).
- You cannot update IMS tables that contain non-unique segments unless both of the following conditions are met:
  - Optimize Update=Y in the IMS Table Definition screen. Refer to [Task D: Select an IMS Database on page 102](#) for more information on Optimize Update.
  - All segments in the access path, starting with the segment where the primary key is defined, are fully defined to TIBCO Object Service Broker.
- REPLACES in IMS tables with overlapping fields are not allowed.
- Updates to child segments of variable length parent segments are not currently supported.

- You can update a variable length segment, provided it is the last segment in the access path.

## Insert Processing

An INSERT request to an IMS table inserts all segments defined in the access path, starting with the segment where the primary key is defined. Parameterized segments are used for positioning only.

All segments in the access path, starting with the segment where the primary key is defined, must have all fields defined to TIBCO Object Service Broker.

INSERTs into IMS tables with overlapping fields are not permitted.

## Delete Processing

A DELETE request against an IMS table deletes either the last TIBCO Object Service Broker IMS occurrence retrieved or the occurrence that contains the primary key value specified in the DELETE request.

The segment the primary key is defined at and *all* child segments associated with that parent are deleted. Parameterized segments are used for positioning only.

You cannot delete occurrences from IMS tables containing non-unique segments unless Optimize Update=Y in the IMS table definition. Refer to [Task D: Select an IMS Database on page 102](#) for more information on Optimize Update.

**See Also** *TIBCO Object Service Broker Parameters* about the TRANMAXNUM Execution Environment parameter.

*TIBCO Object Service Broker Programming in Rules* about rules language statements and coding.

## Understanding Gateway Behavior

---

This section describes how the Gateway handles TIBCO Object Service Broker requests with respect to:

- Synchronization and recovery
- Error handling

### Synchronization and Recovery

Locking of IMS data is determined by the PROCOPT parameter for the database PCB in the Gateway PSB and not by TIBCO Object Service Broker. The IMS unit of work spans the same length of time as a TIBCO Object Service Broker transaction.

DL/I CHKPs are sent from the Gateway to IMS at the end of a transaction. After a COMMIT or ROLLBACK is issued from a rule, an end of transaction is required.

- A COMMIT request sent to the Gateway, or a normal end of transaction results in a DL/I CHKP being sent to IMS.
- A ROLLBACK request sent to the Gateway or a transaction failure results in a DL/I ROLB being sent to IMS.
- Intermediate COMMITs and ROLLBACKs are *not* allowed if IMS data is updated.



The exception COMMITLIMIT does not apply to IMS tables. Requests to update IMS data are processed as they are encountered, and are not buffered in the intent list.

### Updating TIBCO Object Service Broker and IMS and DB2 Data

TIBCO Object Service Broker provides a method of ensuring data integrity when a transaction updates TIBCO Object Service Broker and IMS and DB2 data in the same transaction. This method is referred to as Fail Safe level-1 processing.

If you did not request Fail Safe processing, transactions that update TIBCO Object Service Broker and IMS and DB2 data can result in discrepancies if the Gateway, the Data Object Broker, or the IMS Control Region abnormally terminate during transaction end processing. Refer to [Implementing Fail Safe Processing on page 84](#) for more information.

For IMS data access only, when a ROLLBACK request is sent to the Gateway, a DL/I ROLB is sent to IMS and the TDS intent list is discarded.



For concurrent access to IMS and DB2 data, when a ROLLBACK request is sent to the Gateway, a DL/I ROLB is sent to IMS, IMS coordinates the ROLLBACK with DB2, and the TDS intent list is discarded.

## Gateway for IMS Exceptions

The TIBCO Object Service Broker runtime environment signals system exceptions so an application can recover from an error. A three-level hierarchy of exceptions exists. All errors encountered when accessing IMS or DB2 data through the Gateway are trapped under one of the following TIBCO Object Service Broker exceptions:

|               |  |
|---------------|--|
| ERROR         | An error is detected and no lower-level exception handler exists in the application.                                       |
| ACCESSFAIL    | A table access error is detected.  |
| GETFAIL       | No occurrence satisfies the selection criteria. The IMS status codes GB and GE raise this exception.                       |
| DELETEFAIL    | The primary key specified for a DELETE statement does not exist. The IMS status codes GB and GE raise this exception.      |
| INSERTFAIL    | The primary key provided for an INSERT statement already exists. The IMS status codes II, LB, and LD raise this exception. |
| REPLACEFAIL   | The primary key provided for a REPLACE statement does not exist. The IMS status codes GB and GD raise this exception.      |
| INTEGRITYFAIL | A table access error is detected.  |

|                |   |
|----------------|---|
| DEFINITIONFAIL | A table definition error is detected. The IMS status codes AC, AK, AM, AT, AU, GD, and V1 raise this exception.   |
| LOCKFAIL       | There is a lock on a segment occurrence. The IMS status code BA raises this exception.  |
| SECURITYFAIL   | <p>Permission for the requested action on the TIBCO Object Service Broker object is denied.</p> <p>This can also occur if external security is implemented and:</p> <ul style="list-style-type: none"> <li>• The EXTERNALUSERID parameter is set to GROUP and it is greater than eight (8) characters</li> <li>• One of the checks performed by the System Authorization Facility (SAF) compliant external security package failed</li> </ul> <p>Refer to <a href="#">Implementing External Security on page 76</a> for more information.</p> |
| SERVERBUSY     | A new transaction requested an instance of the Gateway, and no Gateway is available or running. Control is passed back to the rule so it can try the transaction again. If the exception is raised very often, consider more Gateways or review the amount of work being done in your transactions.   |
| SERVERERROR    | The Gateway made a request to IMS, and IMS returned an error code that does not map to an TIBCO Object Service Broker exception. The ON SERVERERROR handler can call @SERVERERROR to parse the error message contained in ENDMSG.   |

|              |  |
|--------------|--|
| SERVERFAIL   | <p>One of the following conditions occurred and control is passed back to the rule for transaction cleanup:</p> <ul style="list-style-type: none"> <li>• The Gateway failed.</li> <li>• A transaction was in progress when the connection to an instance of the Gateway broke.</li> <li>• Integrity with the current SL/I unit of work was lost. IMS backs out all database updates made by the Gateway since the last checkpoint. The IMS status codes BB and BC raise this exception.</li> </ul> |
| VALIDATEFAIL | <p>An attempt was made to update a table with invalid data. The data could have failed a validation rule or reference checking. The IMS status codes DA, DX, IX, NI, and RX raise this exception.</p>  |

See Also     *TIBCO Object Service Broker Programming in Rules* about exception handling.

## Using @SERVERERROR

You must pass [@SERVERERROR](#) the contents of [RETURN\\_MESSAGE](#), which has the following format:

```
pppIMnnnx serverid serveruserid source: Message
```

The following list describes the variables necessary to pass the RETURN\_MESSAGE contents to [@SERVERERROR](#):

|                     |   |
|---------------------|---|
| <i>ppp</i>          | The user-specified 3 character product ID.  |
| <i>nnn</i>          | The IMS external message number.  |
| <i>x</i>            | The message severity (E for error, W for warning, and I for Information).   |
| <i>serverid</i>     | The server ID of the Gateway.   |
| <i>serveruserid</i> | The server user ID (IDPREFIX + ###) of the Gateway.   |
| <i>source</i>       | The code portion of the server that trapped the error and returned the message (for example, CSECT, rule, or function). |
| <i>Message</i>      | The actual error message text.  |

If a specific message from a specific Gateway has some information that is required to process the error, the table-driven approach to the execution of `@SERVERERROR` causes a rule (specified for that error by the developer using `@SERVERERROR`) to execute. The error message is interpreted in the `@SERVERERROR` processing and put into a temporary table until required.

## Customizing Error Handling

To customize error handling, you must update data in one of the control tables `@IMSMSCNTL` or `@IMSMGRCCNTL`. The definition of these tables is owned by TIBCO Object Service Broker and must not be modified. The data you update in them is owned by you.

## Processing Tables

The list below shows how tables are processed when the `SERVERERROR` exception is raised and the `@SERVERERROR` rule is called by your application:

- `@SERVERERROR` reads the `@SERVERMSGCNTL` table and looks up the specific message identifier handlers.
- The appropriate message handler looks up the external error codes in the correct Gateway control tables.
- If any codes are found, they call the associated user-written handler.
- The user-written handler can use other functions and data stored in specific tables to handle any specific external error/status code.

`@SERVERERROR` can be called at any time, although it is useful only for parsing TIBCO Object Service Broker IMS messages generated due to external IMS errors. The original message can always be retrieved using `@SE_MSG` after `@SERVERERROR` has been called. The information parsed by `@SERVERERROR` has transaction scope.

## Adding Instances to the @SERVERMSGCNTL Table

You can add your own instances in the `@SERVERMSGCNTL` table, provided that the OWNER specified begins with letters A to Z, and the key values in their instance are message identifiers in the form `IMnnnnx` mentioned on [page 139](#).

See Also *TIBCO Object Service Broker Shareable Tools* for more information on the `@SERVERERROR` and `RETURN_MESSAGE` tools.

## Appendix A    **Documenting IMS Tables**

This appendix discusses how to document TIBCO Object Service Broker IMS tables.

### Topics

---

- [Using the Documentation Screen, page 142](#)

Each table definition in TIBCO Object Service Broker has a Documentation screen associated with it. You use this screen to create or modify documentation for the table. To display the Documentation screen for an IMS, press PF2 from the Table Definer.

|                      |             |             |
|----------------------|-------------|-------------|
| DESCRIPTION OF TABLE | CUST_ORDERS | UNIT: USR40 |
|----------------------|-------------|-------------|

MODIFIED ON 20 JAN 1998 BY ACC                      CREATED ON 01 JAN 2000 BY USR40

KEYWORDS: ORDER

SUMMARY : IMS TABLE CONTAINING ORDER INFORMATION

DESCRIPTION

— This table contains all information on current customer orders.

PFKEYS: 3=END 5=VIEW DOCUMENT 13=PRINT 12=EXIT

The Table Definer updates some of the fields on this screen. You must maintain the **KEYWORDS**, **SUMMARY**, and **DESCRIPTION** fields:

|                 |   |
|-----------------|---|
| <b>KEYWORDS</b> | Type individual words that briefly describe the table. These words are used by the TIBCO Object Service Broker Keyword Search. This field is one line long and can contain multiple entries, separated by commas or blanks. |
|-----------------|---|

|                    |   |
|--------------------|---|
| <b>SUMMARY</b>     | Type a one line summary of the <b>DESCRIPTION</b> field.  |
| <b>DESCRIPTION</b> | Type information about the table (for example, what its role is, what it does, and how it works) using TIBCO Object Service Broker <b>SCRIPT</b> commands. There is no limit to the amount of information you can type in this field. |

## PF Keys

The following function keys are supported from the Documentation screen:

|      |   |
|------|---|
| PF1  | Displays corresponding help for the current field or screen.  |
| PF3  | Saves changes and returns you to the Table Definer.           |
| PF5  | Toggles between browse and edit modes.                        |
| PF12 | Cancels changes and returns you to the Table Definer.         |
| PF13 | Prints the version of the documentation that you are viewing. |

**See Also** The *TIBCO Object Service Broker Shareable Tools* manual for more information on the [SCRIPT](#) tool.





## Appendix B **Samples of Recommended Table Definitions**

This appendix contains two samples of IMS Table Definitions.

### Topics

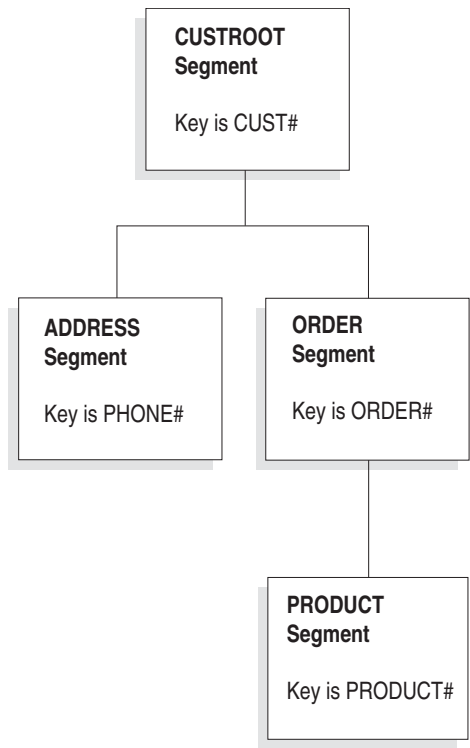
---

- [Sample Table Definitions, page 146](#)

# Sample Table Definitions

## Sample IMS Database with Unique Segments

If you need to directly retrieve or update specific IMS data, you should have one TIBCO Object Service Broker IMS table definition for each segment that can be uniquely identified. The illustration below shows a sample IMS database where all segments contain unique sequence keys.



## Creating the Sample Table Definition

To create an IMS table definition for each segment in the illustration above, complete the following steps:

1. Determine the unique sequence key of each segment in the database using the control table @IMSSEGS(*dbname*).

2. Create a CUSTOMER table that includes the root segment:

---

|          |   |
|----------|---|
| CUSTROOT | Select CUST# as the primary key.<br>Select and define all remaining fields. |
|----------|---|

---

3. Create an ADDRESS table that includes two segments:

---

|          |                                |
|----------|--------------------------------|
| CUSTROOT | Select CUST# as the parameter. |
|----------|--------------------------------|

---

|         |  |
|---------|--|
| ADDRESS | Select PHONE# as the primary key.<br>Select and define all remaining fields. |
|---------|--|

---

4. Create a PURCHASE\_ORDER table that includes two segments:

---

|          |                                |
|----------|--------------------------------|
| CUSTROOT | Select CUST# as the parameter. |
|----------|--------------------------------|

---

|       |  |
|-------|--|
| ORDER | Select ORDER# as the primary key.<br>Select and define all remaining fields. |
|-------|--|

---

5. Create a PRODUCTS\_ORDERED table that includes three segments:

---

|          |                                |
|----------|--------------------------------|
| CUSTROOT | Select CUST# as the parameter. |
|----------|--------------------------------|

---

|       |                                 |
|-------|---------------------------------|
| ORDER | Select ORDER# as the parameter. |
|-------|---------------------------------|

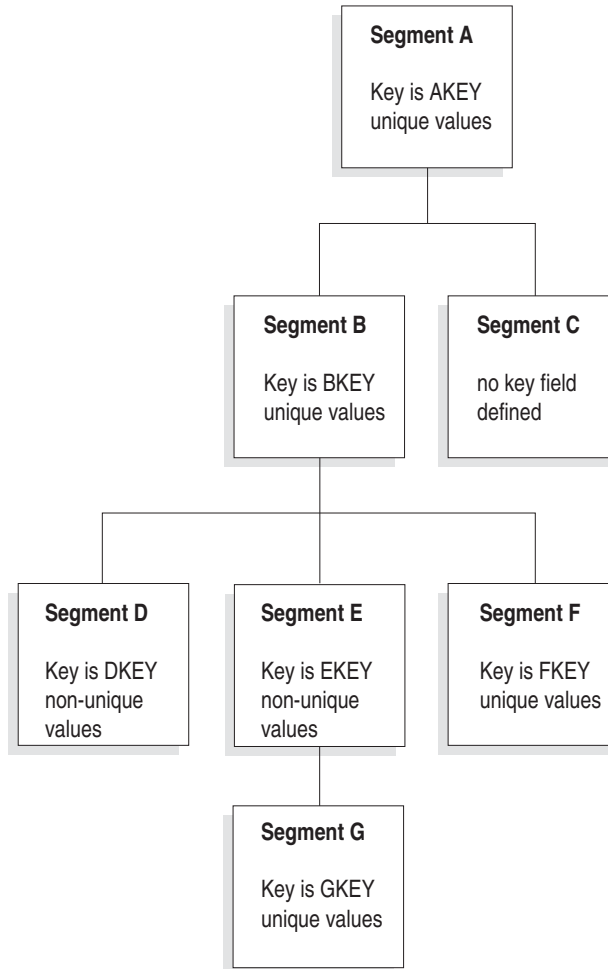
---

|         |  |
|---------|--|
| PRODORD | Select PRODUCT# as the primary key.<br>Select and define all remaining fields. |
|---------|--|

---

## Sample IMS Database with Non-unique Segments

The illustration below shows an IMS database containing non-unique segments.



### Retrieving and Updating Non-unique IMS Data

If you need to directly retrieve or update non-unique IMS data, complete the following steps:

1. Determine the type of key (unique, multiple, or no key) for each segment of the database using the `@IMSSEGS(dbname)` control table.

2. Create TABLE\_A that includes the root segment:

---

A    Select AKEY as the primary key.  
       Select and define all remaining fields.

---

3. Create TABLE\_B that includes two segments:

---

A    Select AKEY as the parameter.

---

B    Select BKEY as the primary key.  
       Select and define all remaining fields.

---

4. Create TABLE\_C that includes two segments:

---

A    Select AKEY as the parameter.

---

C    Select a field as the primary key.  
       Select and define all remaining fields.

---

If you plan to modify IMS data using this definition, specify Optimize Update=Y in the IMS Table Definition screen.

5. Create TABLE\_D that includes three segments:

---

A    Select AKEY as the parameter.

---

B    Select BKEY as the parameter.

---

D    Select DKEY or another field as the primary key.  
       Select and define all remaining fields.

---

If you plan to modify IMS data using this definition, specify Optimize Update=Y in the IMS Table Definition screen.

6. Create TABLE\_E that includes three segments:

|   |   |
|---|---|
| A | Select AKEY as the parameter.   |
| B | Select BKEY as the parameter.   |
| E | Select EKEY or another field as the primary key.<br>Select and define all remaining fields. |

If you plan to modify IMS data using this definition, specify Optimize Update=Y in the IMS Table Definition screen.

7. Create TABLE\_F that includes three segments:

|   |   |
|---|---|
| A | Select AKEY as the parameter.   |
| B | Select BKEY as the parameter.   |
| F | Select FKEY or another field as the primary key.<br>Select and define all remaining fields. |

8. Create TABLE\_G that includes four segments:

|   |   |
|---|---|
| A | Select AKEY as the parameter.   |
| B | Select BKEY as the parameter.   |
| E | Select EKEY or another field as the primary key.<br>Select and define all remaining fields. |
| G | Select and define all fields.   |

Segment G has a unique sequence key; however, if you plan to modify IMS data using this definition, you must still specify Optimize Update=Y in the IMS Table Definition screen, since its parent (segment E) does not have a unique sequence key.



For definitions containing non-unique IMS segments, the Gateway requires a minimum of two identical PCBs defined for the database in question. Refer to [Retrieval Processing on page 132](#) for more information.

# Index

## Symbols

@DB2FSTRXDB file [63](#)  
 @IMSDBD control table [51](#)  
 @IMSDBD table [54](#)  
 @IMSDBS control table [51](#)  
 @IMSFSTRXDB file [63](#)  
 @IMSSEGS control table [51](#)  
 @IMSSIXFIELDS control table [51](#)  
 @STATICSQL rule [56](#)

## A

access paths, determining [107](#)  
 access types [117](#)  
 access values, defining [119](#)  
 ACCESSFAIL exception [137](#)  
 accessing  
   DB2 and IMS data in same transaction [103](#)  
   DB2 data [2](#)  
   IMS data [2](#)  
   IMS fields [72](#)  
   IMS tables [97](#)  
 adding DDnames used by the Service Gateway for  
   IMS/DB [30](#)  
 Administration menu, RESOURCE MANAGEMENT  
   option [48](#)  
 AMODE link control card [29, 56](#)  
 APPLCTN macro [28](#)  
 authorizing  
   access to IMS data [31, 66](#)  
   data sets for Cross Memory Services [28](#)  
   Service Gateway for IMS/DB, for SAF [77](#)

## B

binding IMS table definitions [55](#)  
 Browse PSBname field, table definition screen [105](#)  
 browsing IMS data [128](#)

## C

CANCEL primary command [100](#)  
 CHANGE\_SERVERID tool [90, 90](#)  
 child segments, displaying [107](#)  
 CICS environment, preparing [30](#)  
 CICS Processing Program Table, modifying [30](#)  
 CICSHURONTRAN parameter [30, 57](#)  
 CLASS parameter [31, 59](#)  
 CMPAT option [27](#)  
 COMMIT [136](#)  
 COMMITLIMIT exception [136](#)  
 communications [5](#)  
 communications from TIBCO Object Service Broker to  
   IMS [4](#)  
 consistency of data [84](#)  
 COPY primary command [99](#)  
 copying TIBCO Object Service Broker objects,  
   tools [101](#)  
 Cross Memory Services  
   authorizing data sets for [28](#)  
   usage [5](#)  
 CTABLESIZE parameter [64, 72, 114, 131](#)  
 customer support [xvi](#)  
 customizing  
   OSEMOD IMS and DB2 variables [6](#)  
 customizing the Program Control Table [30](#)

## D

- D (delete field) line command 115
- data integrity 84, 136
- data mapping considerations 123
- Data Object Broker 5, 92
- data type mapping 121
- database-level security, establishing 79
- DB Name field, header segment 110
- DB NAMES primary command 99
- DB2 access
  - external security interface 32
  - linking the static SQL handlers 29
  - plan name 64
  - preparing 56
- @DB2FSTRXDB file 63
- DBD load libraries 50
- DBD load modules 50
- DBDCLASS parameter 32, 60
- DEBUG parameter 60, 91
- debugging rules 90
- Dec field, Metadata Field section 112
- DECOUPL parameter 60
- dedicated Gateway 90, 90
- defining
  - access values 119
  - DB2 transaction table 85
  - event rules 117
  - Gateway program specification blocks (PSBs) 26
  - Gateway to the IMS Online system 28
  - IMS Secondary Index access 27
  - IMS tables 96, 97
  - IMS transaction database 85
  - multiple record formats 120
- DELETE
  - primary command 100
  - statement 135
- delete processing 135
- deployment 5
- DESCRIPTION field 143
- determining
  - access paths 107
  - SQL statements 90
- DISPLAY & TRANSFERCALL statement 131
- display order, modifying 115

- Distributed Data 86
- distribution file format 7
- DL/I Batch Gateway, starting 40
- DL/I CHKP call 136
- DL/I ROLB call 136
- DL/I Trace Facility 91
- DLIQUERYLIMIT parameter 61
- DLIUPDATELIMIT parameter 60
- DOC primary command 99
- Documentation screen PF keys 143
- documenting IMS tables 141
- domain requirements 5
- DRASUFFIX parameter 57

## E

- editing IMS data 128
- ERROR exception 137
- establishing
  - database-level security 79
  - PSB-level security 78
  - resource- and database-level security 80
  - resource- and segment-level security 81
  - resource-level security 77
  - segment-level security 80
- ESTIMATETBLDFN rule 72, 114
- Event Rule segment, table definition screen 99
- event rules, defining 117
- exception handling 137
- EXECUTE statement 131
- external security
  - database-level 77
  - implementing 76
  - methods 76
  - PSB-level 77
  - resource-level 77
  - segment-level 77
- EXTERNALGROUP parameter 61, 76
- EXTERNALUSERID parameter 62, 76
- extracting
  - IMS database definitions 96
  - IMS database information 50



- extracting IMS database information
  - sample JCL 50
  - sample SYSIN member 50

## F

- Fail Safe level-1 processing
  - and transaction updates 136
  - implementing 84
  - modifying parameters 85
- Fail Safe processing, activating 62
- Fields segment, table definition screen 99
- fields, selecting IMS 109
- FORALL statement 133
- FSLEVEL parameter 38, 85
- FSLEVEL the Gateway parameter 62
- FSTABLENAME Gateway parameter 63, 85
- full definition, location parameter 106

## G

- Gateway configuration requirements 5
- Gateway number allowed 131
- Gateway pool 67
- Gateway repository file 37
- Gateway status, displaying 86
- Get Hold Unique 129
- GET statement 133
- GET...ORDERED statement 133
- GHU. *See* Get Hold Unique 129

## H

- HDAM databases 132
- Header segment, table definition screen 98
- Help, displaying 99

## I

- I (insert field) line command 115
- IDPREFIX parameter 63
- Idx field, IMS field section 111
- implementing
  - external security 76
  - TIBCO Object Service Broker security 31
- IMS data
  - browsing 128
  - editing 128
  - TIBCO Object Service Broker interface to 2
- IMS data types, conversion to TIBCO Object Service
  - Broker syntax 121
- IMS database access, verifying 31
- IMS database segment, table definition screen 98
- IMS databases
  - extracting information 50
  - importing database definitions 51
  - selecting 102
- IMS environment, preparing 26
- IMS Field Selection screen 109
- IMS fields
  - associating an access value 115
  - data type 112
  - selecting 114
  - syntax 111
  - using as a parameter 115
  - using as primary keys 114
- IMS logical database 27
- IMS offsets 111
- IMS Online system, defining the Gateway to 28
- IMS processing
  - effect of field selection on 124
  - effect of segment selection on 123
- IMS resources, verifying access to 76
- IMS Secondary Index Access, defining 27
- IMS segment, table definition screen 99
- IMS segments, selecting 107
- IMS subsystem attach facility 2
- IMS table definitions
  - binding 55
  - maximum space for 55
  - rebinding 55

## IMS tables

- accessing 97
- defining 97
- documenting 141
- restricting ability to define 31

## IMS tool 51

## IMS transaction database, defining 85

## @IMSFSTRXDB file 63

## @IMSDBD control table 51

## @IMSDBS control table 51

## IMSfields, deleting 115

## IMSGEN table 51

## @IMSSEGS control table 51

## @IMSSIXFIELDS control table 51

## IMSSRV00 DDname 30

## INDICES parameter 27, 115

## insert processing 135

## installation

- media 7, 12
- receiving initial file 9

## installation variables, OSEMOD 6

## installing Service Gateway for IMS/DB 9, 11

## INTEGRITYFAIL exception 137

## intent lists, and IMS data 136

## interface to IMS data 1, 2

## intermediate COMMITs 136

## intermediate ROLLBACKs 136

**K**

## K (primary key) line command 114

## KEYWORDS field 142

**L**

## LANG= option 27

## Len field, IMS field section 111

## Len field, Metadata Field section 112

## level of Fail Safe processing 62

## levels of system exceptions 137

## loading IMS database definitions 51

## Location Parameter segment, table definition

- screen 98

## location parameters

- defining 106
- using 96

## locking of IMS data 129, 133, 136

## logical database definition, completing 53

## logical database, creating 53

**M**

## Max Length field, header segment 110

## maximum space for IMS table definitions 55

## maximum transaction streams 131

## MDL parameter 57

## Min Length field, header segment 110

## minimal definition 106

## MODE parameter 64

## MODIFY operator command 45, 47, 47

## modifying

- CICS Processing Program Table 30

- Fail Safe startup parameters 85

- server IDs 90

## multiple record formats, defining 120

## MVS installation 9

**N**

## non-unique IMS segments

- accessing 134
- retrieving 134

**O**

## Object Manager screen 97

## obtaining installation media 7, 12

## Offline DL/I Batch Processing Region 26

## Offset field, IMS field section 111

## operating systems 5

Optimize Update field, table definition screen 104  
 Ord field, Metadata Field section 113  
 Order field 115  
 OSEMOD installation variables 6  
 OTMA 2

## P

P (parameter) line command 115  
 Parameter segment, table definition screen 99  
 PATH calls 26, 132  
 PCB for transaction database, defining 85  
 PCB. *See* Program Control Block  
 PCT. *See* Program Control Table  
 PF keys  
   Documentation screen 143  
   primary commands, listed 99  
   Table Definition screen 99  
 PF Keys segment, table definition screen 99  
 PLAN parameter 64  
 pool of Gateway 67  
 POOLSIZE parameter 55, 64, 131  
 PPT. *See* CICS Processing Program Table  
 preparing  
   CICS environment 30  
   IMS environment 26  
   Offline DL/I Batch Processing Region 28  
 preparing DB2 access 56  
 primary authorization ID for recovering 65  
 primary commands, listed 99  
 primary keys, specifying 114  
 PRINT primary command 100  
 problem reporting 91  
 PROCOPT parameter  
   Gateway PSB requirements 26  
   locking IMS data 129  
 PROCSEQ parameter 27, 115  
 Program Control Block 104  
 Program Control Table, customizing 30  
 Program Specification Block, defining 26  
 PSB. *See* Program Specification Block  
 PSBCLASS parameter 64  
 PSB-level security, establishing 78

PSBname field, table definition screen 105  
 PSBNAME parameter 64

## R

rebinding IMS table definitions 55  
 receiving initial installation file 9  
 recovery 136  
 RECOVERYID parameter 65  
 replace processing 134  
 REPLACE statement 134  
 reporting problems 91  
 REQSTOR parameter 65  
 request handling 136  
 RESETXPARM tool 74  
 resource- and database-level security, establishing 80  
 resource- and segment-level security, establishing 81  
 RESOURCE MANAGEMENT option  
   description 48  
   displaying Gateway status 86  
 RESOURCE parameter 31, 65  
 resource repository file 37  
 resource-level security, establishing 77  
 RESPONSEMODE parameter 38, 65  
 restricting ability to define IMS tables 31  
 retrieval processing 132  
 RMODE link control card 29, 56  
 ROLLBACK 136  
 Rqd field, Metadata Field section 113  
 Rule Debugger 90  
 rules, using to access IMS data 131

## S

S (select field) line command 114  
 S6BCSDLI program 30  
 S6BIMSU program 50, 115  
 SAVE primary command 99  
 SCOPE parameter 66  
 SECLEVEL parameter 31, 66, 76  
 secondary index, using to access IMS data 53

- security, implementing 31
- SEGCLASS parameter 67
- Segment field, header segment 110
- Segment Name field 107
- Segment Search Argument 128
- segment-level security, establishing 80
- selecting
  - IMS databases 102
  - IMS fields 109
  - IMS segments 107
- server configuration requirements 92
- Server ID field, table definition screen 103
- server ID, modifying 90
- Server Type field, table definition screen 103
- SERVERID parameter 39, 41, 67, 90, 90, 103
- SERVERPARM parameter 58
- SERVERS parameter 58
- SERVERTYPE parameter 58
- Service Gateway for IMB/DB, installing 11
- Service Gateway for IMS/DB
  - adding DDnames 30
  - authorizing for SAF 77
  - CICS Execution Environment 41, 42
  - exceptions 137
  - installing 9
  - installing on a remote host 12
  - interfaces supported by 2
  - operation 136
  - OSEMOD installation variables 6
  - overview 2
  - parameters 34
  - running 3, 39
  - running APF authorized 77
  - status, displaying 86
- SETXPARM tool 74
- Single Occurrence Editor, using to access IMS
  - data 129
- space for TIBCO Object Service Broker IMS database
  - definitions 64
- SQL statements, determining 90
- SSA. *See* Segment Search Argument 128
- starting TIBCO Service Gateway for IMS/DB 39–44
- startup prerequisites 37
- Static SQL handlers 56
- @STATICSQL rule 56

- STEPLIB data set 28
- Substation ES, MVS installation procedure 9
- SUBSYS parameter 67
- SUMMARY field 143
- support, contacting xvi
- Syn field
  - IMS field section 111
  - Metadata Field section 112
- synchronization 136
- system exceptions 137

## T

- Table Browser, using to access IMS data 128
- Table Definition screen
  - PF keys 99
  - segments 98
- Table Editor, using to access IMS data 128
- Table field, header segment 101, 110
- TDS parameter 59
- technical support xvi
- TIBCO Object Service Broker CICS transaction
  - identifier 30
- TIBCO Object Service Broker interface to IMS data 1, 2
- TIBCO Object Service Broker objects, copying 101
- TIBCO Service Gateway for IMS/DB, starting 39–44
- TIBCO\_HOME xiii
- tools
  - CHANGE\_SERVERID 90
  - RESETXPARM 74
  - SETXPARM 74
- TRANMAXNUM parameter 131
- transaction streams 131
- transaction updates, recovering 136
- TRANSFERCALL statement 131
- trigger rules 117
- TRXDB Gateway parameter 85
- type
  - event rule 117
  - of access 117

## Type field

- header segment [101, 110](#)
- Metadata Field section [112](#)
- table definition screen [98](#)

## U

- unique sequence key fields [134](#)
- unique sequence keys [129](#)
- Unit field, header segment [101, 110](#)
- update processing. *See* replace processing
- uploading the software [8](#)
- user ID, controlling identification to IMS [62](#)
- USERTYPE parameter [38, 41, 42, 44, 70](#)

## V

- V (access value) line command [115](#)
- Val field, IMS field section [112](#)
- validation rules [117](#)
- VERIFYEXTACCESS parameter [71](#)
- verifying
  - access to IMS resources [76](#)
  - IMS database access [31](#)
- VTAM ACB name used for communication [57](#)
- VTAM applid of Data Object Broker [59](#)

## X

- Xsyn field, IMS field section [112](#)