TIBCO® Object Service Broker

Managing Security

Software Release 6.0
July 2012

two-second advantage™

TIBCO®
The Power of Now®

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

Contents

Preface

TIBCO® Object Service Broker is an application development environment and integration broker that bridges legacy and non-legacy applications and data.

This manual describes how to set up, use, and administer the security required for a TIBCO Object Service Broker development environment.

Topics

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
TIBCO_HOME *OSB_HOME*	By default, all TIBCO products are installed into a folder referenced in the documentation as *TIBCO_HOME*. On open systems, TIBCO Object Service Broker installs by default into a directory within *TIBCO_HOME*. This directory is referenced in documentation as *OSB_HOME*. The default value of *OSB_HOME* 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 *TIBCO_HOME*. For example on Windows systems, the default value is `C:\tibco\OSBgateways\6.0`. On z/OS, no default installation directories exist.
`code font`	Code font identifies commands, code examples, filenames, pathnames, and output displayed in a command window. For example: Use `MyCommand` to start the foo process.
`bold code font`	Bold code font is used in the following ways: • In procedures, to indicate what a user types. For example: Type **`admin`**. • In large code samples, to indicate the parts of the sample that are of particular interest. • In command syntax, to indicate the default parameter for a command. For example, if no parameter is specified, `MyCommand` is enabled: `MyCommand [`**`enable`**` | disable]`
italic font	Italic font is used in the following ways: • To indicate a document title. For example: See *TIBCO ActiveMatrix BusinessWorks Concepts*. • To introduce new terms For example: A portal page may contain several portlets. *Portlets* are mini-applications that run in a portal. • To indicate a variable in a command or code syntax that you must replace. For example: `MyCommand` *PathName*

Table 1 General Typographical Conventions (Cont'd)

Convention	Use
Key combinations	Key name separated by a plus sign indicate keys pressed simultaneously. For example: Ctrl+C. Key names separated by a comma and space indicate keys pressed one after the other. For example: Esc, Ctrl+Q.
	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
`[ ]`	An optional item in a command or code syntax. For example: `MyCommand [optional_parameter] required_parameter`
`|`	A logical `OR` that separates multiple items of which only one may be chosen. For example, you can select only one of the following parameters: `MyCommand para1 | param2 | param3`

Table 2 Syntax Typographical Conventions

Convention	Use

`{ }`	A logical group of items in a command. Other syntax notations may appear within each logical group.
	
	For example, the following command requires two parameters, which can be either the pair `param1` and `param2`, or the pair `param3` and `param4`.
	
	`MyCommand {param1 param2} | {param3 param4}`
	
	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`:
	
	`MyCommand {param1 | param2} {param3 | param4}`
	
	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`.
	
	`MyCommand param1 [param2] {param3 | param4}`

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

Aspects of TIBCO Object Service Broker Security

This chapter provides an overview of the different aspects of TIBCO Object Service Broker security.

Topics

What is TIBCO Object Service Broker Security?

TIBCO Object Service Broker provides security management tools to manage security-related issues within the TIBCO Object Service Broker executing environment. These tools provide:

- Login authentication
- User login profiles
- Hierarchical classification levels for users and objects
- Mandatory levels of protection for TIBCO Object Service Broker objects
- Discretionary levels of protection for TIBCO Object Service Broker objects, used in conjunction with the mandatory levels of protection
- An audit of accesses to objects
- Interfaces to other security products

What Tools are Available?

The following table lists the tools that are available to you and where you can get additional information about each of the tools:

Tool	Refer to …
Security Manager—this is the primary tool for TIBCO Object Service Broker security	Chapter 1, Aspects of TIBCO Object Service Broker Security, page 1, Chapter 2, Security Clearances, page 11, Chapter 3, Managing User Profiles, page 25, Chapter 6, Managing Permissions to Objects, page 59, and Chapter 10, Bound Security Access Data, page 131.
Audit log	Chapter 8, Auditing Accesses, page 103.
Archive audit log tools	Chapter 9, Archiving the Audit Log Data, page 119.
Password encryption exit	Chapter 11, Password Encryption API, page 135.
External security interface	Chapter 12, Implementing External Security, page 149.
Rebinding tool for security data	Chapter 10, Bound Security Access Data, page 131.
Security statistics analysis tool	*TIBCO Object Service Broker for z/OS Monitoring Performance*

How Does TIBCO Object Service Broker Security Relate to External Security?

TIBCO Object Service Broker security encompasses TIBCO Object Service Broker sessions and their Execution Environment. It determines the objects that a user can access from the Pagestore and what types of access a user can make.

External security is used to control access to the Data Object Broker. Depending on the configuration of your Execution Environment, you can also use your external security product to validate access to an Execution Environment or to work in conjunction with TIBCO Object Service Broker security. For a detailed explanation, refer to Login Authentication on page 12 and Chapter 12, Implementing External Security, on page 149.

The Security Manager Facility

What Can the Security Manager Facility be Used For?

You use the Security Manager to:

- Define different types of users

  Your user ID definition determines your use of this interface.

- Set access permissions on the following types of objects: libraries, screens, reports, tables, and object sets

  Access permissions that you specify for a particular object depend upon its object type. Refer to Chapter 6, Managing Permissions to Objects, on page 59 for a description of the allowed accesses for each object type.

User Functions

If you are a user without security administration duties, you use the Security Manager when you want to:

- Modify your user profile, for example, to specify print options or change your password
- Set up a group of users planned to have common security accesses
- Allow other users various levels of access to objects that you created
- Protect objects that you created

Refer to Chapter 3, Managing User Profiles, on page 25, Chapter 5, Managing Security Groups, on page 51, and Chapter 6, Managing Permissions to Objects, on page 59 for detailed information about these uses.

Administrator Functions

If you are an administrator, you use the Security Manager when you want to:

- Perform one of the functions described in User Functions above
- Add or delete user IDs
- Manage the security profiles for other security administrators

In addition to the references above, refer to Chapter 4, Managing Security Administrator Profiles, on page 43.

Accessing the Security Manager

How Do You Access the Interface?

You manage permissions to existing objects from the Security Management main menu. To invoke this menu, do one of the following from the workbench:

- Position your cursor beside the Security Manager option and press Enter:

  ```
  SE Security Mgr ==>
  ```

- Set the workbench Browse flag to Y, enter the tool name, "SECURITY", next to the Execute Rule option, and press Enter:

  ```
  EX execute rule ==> SECURITY
  ```

Security Management Main Menu

Using either option causes the Security Management main menu, shown below, to appear. Use PF1 to display help information for the fields on this screen.

```
--------------------------------------------------------------------------------
                        SECURITY MANAGEMENT MAIN MENU
--------------------------------------------------------------------------------
 MANAGE PERMISSIONS TO OBJECTS

                           Object:                    ( Type:     )
                        ObjectSet:

 SPECIFY PERMISSIONS LISTS

              Default Permissions:
            ObjectSet Permissions:

 MANAGE USERS

                    User Profile:
               Transfer Ownership:
                   Security Group:
                  SecAdmin Profile:

            < Position cursor to desired option; fill in NAME. >

 PFKEYS: 1=HELP ENTER=SELECT 2=OPTIONS 3=EXIT 12=EXIT
```

Methods Available to Enter Values

The following two methods are available to enter values on the menu. In both cases, you must first position your cursor next to the option that you are going to use:

- If you know the value, you can type it in directly next to the option field.
- If you do not know the value, press PF2 to display a list of values, use the **s** line command to select from the displayed list, and press PF3. The value is returned to the option field.

After entering your values in the selected option, use Enter to display the input screen that you require.

Security Roles within the Security Manager

Types of Security Profiles

There are three types of profiles known to the Security Manager:

- Users and groups of users—including end users and application developers
- Security administrators
- System administrators

Each user type has a specific security profile type defined within TIBCO Object Service Broker through the Security Manager facility. Additional individuals could have security administrative duties within your operating environment but outside of TIBCO Object Service Broker security. Their security profiles are defined using an external security package.

User Profile

Each user session is identified by a user ID specified via a user profile. Although a user can have multiple user IDs, only one user ID can be used at session login. Each user is known to a TIBCO Object Service Broker session by the user ID they use at login.

As a user you automatically become the owner of an object when you create it. As its owner, you are permitted all accesses to the object.

You use the Security Manager to grant others access to your objects. Similarly, other users can grant or restrict your user ID access to their objects. As a user, you are known to the Security Manager as an individual user ID and also as a member of one or more security groups.

Security Group Profile

A security group is a list of users who require the same access rights or permissions. A user operates out of only one security group at a time. One user normally belongs to several security groups. For example, the USR01 user ID could belong to the groups listed in the following table:

Security Group	Description
ALL	Every user belongs to the virtual group named ALL. Even if your user ID and your current group are not explicitly permitted access to an object, any permissions for ALL are allowed.
APDEV	Lists all members in USR01's application development team. Provided they operate out of this security group, developers working with USR01 can share access to objects involved in their common application.
SYSTEMS	Lists all members in all application development teams. When they operate out of this security group, all developers are allowed READ access to objects they share.

Accessing Objects

When access is attempted on an object, the Security Manager looks up predefined access permissions for the user ID. This includes the security group that the user ID currently belongs to and the virtual group ALL, to which all user IDs always belong. Summary of Checks When Accessing Objects on page 22 contains information on how security clearance is accomplished when you are accessing an object.

Your user profile specifies your default security group (for example, APDEV). If you also work on other projects, you can change this default by specifying another security group to which you belong as your current group. For more information about setting your current group, refer to Full Name, Phone, Timezone, CURRENT GROUP, and SecAdmin Fields on page 29.

Security Administrator Profile

A security administrator is a user with special privileges within the Security system. Only a system administrator can appoint security administrators.

If you want to access an object and access is denied, the first person you should contact is your security administrator. You can view your user profile to find out who your administrator is. For information about viewing your user profile, refer to Accessing a Profile on page 27.

Privileges

The user IDs of normal users are assigned to a security administrator for administration purposes. These user IDs are known as subjects. Security administrators have the following privileges over their subjects:

- Management of object permissions to all objects owned by their subjects
- Ability to transfer ownership of objects owned by their subjects
- Ability to view, modify, and delete objects owned by their subjects
- Management of memberships to security groups created by their subjects
- Ability to transfer their subjects to another security administrator
- Ability to view the user profiles of their subjects and to view security administrator profiles of other security administrators

Optional Privileges

In addition to the privileges described above, security administrators can optionally be given the following mutually exclusive privileges:

- Create user profiles (the new user ID automatically becomes a subject)
- Modify the user profile of any of their subjects
- Delete the user profile of any of their subjects

System Administrator Profile

A system administrator is a user ID with the highest-level security clearance, level 7. This clearance can be given only to a user ID by another system administrator. Because of the absolute control system administrators have on a TIBCO Object Service Broker system, great care must be taken in granting level-7 clearance and in its use.

Privileges

As a level-7 user, the system administrator has the following privileges:

- Full control over an entire TIBCO Object Service Broker system

This includes creating, changing, and deleting user profiles, and appointing security administrators and other system administrators.

- Full access to all objects in the system, regardless of their owners
- Ability to grant or revoke access privileges for all users and groups
- Ability to perform all security administrator functions

All functions in this manual that belong to a security administrator can also be done by a system administrator.

Chapter 2 Security Clearances

This chapter describes the security clearances that you need to log into a session of TIBCO Object Service Broker and to access any object.

Topics

Login Clearance

Login Authentication

Before you can log in to a TIBCO Object Service Broker session, you must have security login clearance to access TIBCO Object Service Broker. This login clearance is specified through user IDs and passwords. A user ID can be specific to an individual user, or in the case of z/OS it can be a terminal ID or transaction ID. In all cases, to activate a session, each user ID used to access TIBCO Object Service Broker must be:

1. Defined to TIBCO Object Service Broker
2. Authenticated by TIBCO Object Service Broker or an authorized external security process

Types of Login Authentication

The SECURITY Execution Environment parameter supports three authentication policies:

- **INTERNAL** TIBCO Object Service Broker security (see Authentication Using TIBCO Object Service Broker Security on page 13). This is the default.
- **EXTERNAL** External security, such as RACF or Generic Security Service (GSS) (see Authentication Using External Security on page 14).
- **MIXED** Mixed security (see Authentication Using Mixed Security on page 14). TIBCO Object Service Broker security is used first. External security is used if the first verification attempt fails.

The specified policy for your site determines whether the user ID to be associated with your session is either inherited from your external environment or explicitly specified using the USERID=*userid* session parameter.

On z/OS only, you set the SECURITY parameter using the JCL in the EECONFIG member of the CNTL data set distributed with TIBCO Object Service Broker. EECONFIG contains JCL for creating parameter modules for all TIBCO Object Service Broker z/OS environments.

Additional Information

Refer to Chapter 12, Implementing External Security, on page 149 for implementation details if an external package is to be used.

Case Sensitivity of Passwords

In Open Systems, passwords are case sensitive.

On z/OS, passwords case sensitivity is as follow:

- If the Userid profile is set with **Ext Security Mixed-case Password: N** and you type your password on z/OS, TIBCO Object Service Broker changes it to uppercase.
- If the Userid profile is set with **EXT Security Mixed-case Password: Y,** The case sensitivity depends on your external security software. Refer to Ext Security Mixed-case Password on page 31

In a distributed data environment, where Open Systems passwords could be used to access data on z/OS, TIBCO Object Service Broker does not change them to uppercase. In this situation, your Windows or Solaris password must be defined in uppercase.

See Also *TIBCO Object Service Broker Parameters* for a detailed explanation of the Execution Environment and session parameters.

TIBCO Object Service Broker for z/OS Installing and Operating for information about EECONFIG.

Authentication Using TIBCO Object Service Broker Security

User ID and Password Authentication

To start the session, your client session must explicitly supply the password to be associated with your user ID as a session parameter. Your supplied password is validated against the password maintained by TIBCO Object Service Broker security for your user ID.

Interactive Sessions and Password Validation

In interactive sessions, you are prompted for the password if it is not supplied or if the initial password you supplied does not validate.

See Also *TIBCO Object Service Broker Parameters* for information about parameters.

Authentication Using External Security

User ID Authentication

The user ID initiating your session must be authenticated by the external security manager before your session is started with that user ID. The type of session determines the authentication policy.

Seamless and Non-seamless z/OS Clients

Seamless z/OS clients started in CICS or IMS TM external environments are assumed to be authenticated, since the environment is responsible for its own security.

For non-seamless z/OS clients, your user ID is assumed to be authenticated if your TIBCO Object Service Broker user ID is inherited from the external environment. If your user ID is specified by the USERID session parameter (and is different from that which would have been inherited from the external environment), TIBCO Object Service Broker uses the external security manager to authenticate your user ID with the supplied password.

Open Systems Clients

For Open Systems clients, your user ID must be specified by the USERID session parameter. TIBCO Object Service Broker then uses the external security manager to authenticate your user ID with the supplied password.

Interactive Clients and Password Validation

For interactive sessions, you are prompted for the password if the external security manager rejects the supplied password or if you did not supply one.

See Also *TIBCO Object Service Broker Parameters* for information about parameters.

TIBCO Object Service Broker for z/OS External Environments about seamless and non-seamless clients.

Authentication Using Mixed Security

User ID Authentication

If the client is seamless (on z/OS only) or the USERID parameter is not specified, the TIBCO Object Service Broker user ID comes from the external environment. In this case, the user ID is automatically authenticated.

If the client is not seamless (on z/OS) and the USERID parameter is specified, a mixed password validation strategy is used, and the following applies.

Null Password in TIBCO Object Service Broker

If the TIBCO Object Service Broker security system has a null password for your user ID—only permissible on the z/OS environment—and your user ID is explicitly specified by the USERID parameter, the external security manager is used to authenticate your user ID with the supplied password.

For interactive clients, you are prompted for the password if it is not supplied or if the previously supplied password was rejected by the external security manager.

Password in TIBCO Object Service Broker

If the TIBCO Object Service Broker security system has a password for your user ID and your user ID is explicitly specified by the USERID parameter, the supplied password is validated in the following sequence:

1. Against your password maintained by TIBCO Object Service Broker security
2. If that fails, against the password returned from the Version Mismatch function in the password encryption exit

   This function is described in Chapter 11, Password Encryption API, on page 135.
3. If that fails, by the external security manager

   For interactive clients, you are prompted for the password if it is not supplied or if the previously supplied password was rejected by the external security manager.

See Also *TIBCO Object Service Broker Parameters* for information about parameters.

TIBCO Object Service Broker Security—Obtaining and Evaluating User IDs and Passwords

Order of Evaluation and Sources for Values

If your environment uses TIBCO Object Service Broker security only, the user ID and password values for your session can be determined from a number of places. This table describes the possible sources for these values and the order in which they are evaluated:

Order of Evaluation	Source	Platform	Notes
1	Session startup string	z/OS	The client type determines how these values are passed in.
		Open Systems	Passed in via the osBatch utility.
2	ostty parameter file	Open Systems	tty.prm file.
3	Session parameter input file	z/OS	The parameter input file must be allocated to DDname HRNIN. For batch and TSO only.
		Open Systems	session.prm file.
4	Windows Registry	Windows	
5	User profile	z/OS, Open Systems	Used only if the PROFILE/NOPROFILE session option is set to PROFILE. Used for security group user runs under.
6	Execution Environment startup string	z/OS	Specified in the command line region startup string. For batch and TSO only.
7	Execution Environment parameter input file	z/OS	Parameter input file must be allocated to DDName HRNIN. For batch and TSO only.
8	Installation default	z/OS	The default module is determined by the type of Execution Environment.

Order of Evaluation	Source	Platform	Notes
9	User prompts	z/OS	All except Batch, seamless clients, and Call Level Interface.
		Open Systems	ostty prompt screen.
10	Default supplied by TIBCO Object Service Broker	z/OS, Open Systems	

See Also

- *TIBCO Object Service Broker for z/OS External Environments* about specifying the session startup string for each z/OS client type.
- *TIBCO Object Service Broker for Open Systems External Environments* about osBatch.
- *TIBCO Object Service Broker for z/OS Installing and Operating* for more information about HRNIN, about installation, and about defaults supplied by TIBCO Object Service Broker.

Summary of User ID and Password Requirements

The following tables summarize the user ID and password requirements for each of the TIBCO Object Service Broker, external, and mixed authentication policies. The columns of these table are arranged as follows:

- The User ID Inherited column indicates whether the TIBCO Object Service Broker user ID is derived from the external environment (Y) or specified by the USERID=*userid* (N) session parameter.
- The Password Specified column indicates whether a password is specified by the session parameter or as a result of a user prompt for a password.
- The TIBCO Object Service Broker Password Exists column indicates whether TIBCO Object Service Broker security has a password stored for the user ID.
- The Password Matches TIBCO Object Service Broker Password column indicates whether the password provided by the session parameter, or as a result of a prompt, matches the password maintained by TIBCO Object Service Broker security.
- The Password Matches External Security column indicates whether the password provided by the session parameter, or as a result of a prompt, is validated by the external security manager.

TIBCO Object Service Broker Security

User ID Inherited (z/OS only)	TIBCO Object Service Broker Password Exists	Password Specified	Password Matches TIBCO Object Service Broker Password	Result
Not applicable	Y	Y	Y	Login
Not applicable	Y	Y	N	Prompt [a, b]
Not applicable	Y	N	---	Prompt [c]
Not applicable	N	---	---	Reject

a. Try the password from the encryption exit mismatch function, if applicable.

b. Reject if not an interactive session or more than three prompts.

c. Reject if not an interactive session.

External Security

User ID Inherited (z/OS only)	Password Specified	Password Matches External Security	Result
Y	---	---	Login [a]
N	Y	Y	Login
N	Y	N	Prompt [b]
N	N	---	Prompt [c]

a. A specified password is ignored.

b. Reject if not an interactive session or more than three prompts.

c. Reject if not an interactive session.

Mixed Security

User ID Inherited (z/OS only)	Password Specified	TIBCO Object Service Broker Password Exists	Password Matches TIBCO Object Service Broker Password	Password Matches External Security	Result
Y	---	---	---	---	Login [a]
N	Y	Y	Y	---	Login
N	Y	Y	N	Y	Login [b]
N	Y	Y	N	N	Prompt [b, c]
N	Y	N	---	Y	Login
N	Y	N	---	N	Prompt [c]
N	N	---	---	---	Prompt [d]

a. A specified password is ignored.

b. The encryption exit mismatch function is always called if the specified password does not match the TIBCO Object Service Broker password.

c. Reject if not an interactive session or more than three prompts.

d. Reject if not an interactive session.

Object-Level Clearance

Clearance Checks When Accessing Objects

Evaluating Accesses

Whenever you attempt to access an object, the Security Manager allows or disallows the access, subject to the following sequence of clearance checks:

1. The security clearance level of the user ID you are using to access the object must be equal to or greater than the classification level of the object being accessed.
2. The user ID must have permission for the particular access type that is being attempted on the object.

These values are set either through the Security Manager facility or through default by TIBCO Object Service Broker security.

Supported Clearance Levels

From within the Security Manager, a security administrator assigns a clearance level when defining your user IDs to TIBCO Object Service Broker. This clearance level, which is hierarchically evaluated, is used for each TIBCO Object Service Broker session initiated by your user ID. Only three clearance levels are supported:

Clearance Level	Description
7	The highest clearance level. It defines a user ID as a system administrator.
1	The normal user ID level of clearance. It is used for developers, end-users, and for most security administrators.
0	The lowest clearance level. It indicates that the user ID belongs to a suspended user.

Classification Levels

When you create an object, the object is assigned the clearance level of your creating user ID. This is known as a mandatory classification level. If you are the owner of the object, you can change this classification level for the object. For more information about changing classification levels, refer to Task C: Modify the Classification Level on page 69.

If your user ID has a clearance level of 1, you cannot access an object with a classification level of 7. A user ID with a clearance level of 7 can modify an object with a classification level of 1 and the modified object maintains its classification level of 1.

Discretionary Access Permissions

After creating an object, you can assign permissions for particular types of access to the object. For example, you could assign other users READ access to the object but not INSERT or REPLACE access. These discretionary permissions, which are non-hierarchical, are then evaluated each time an access is made to the object.

You can grant a user ID permissions to an object by:

- Explicitly giving permissions to the user IDs
- Explicitly giving permissions to a group to which the user ID belongs
- Making the object part of an object set that the user ID can access

For more information about granting permissions to objects, refer to Chapter 6, Managing Permissions to Objects, on page 59, and for more information about object sets, refer to Chapter 7, Managing Object Set Security, on page 87.

Access Permissions to Object Sets

Access permissions are properties of individual objects. As part of defining the composition of an object set, you indicate the access permissions for each individual object in the object set. When you give a user ID or a security group access to an object set, you are allowing access to all objects in the set, based on the accesses specified for each object. Refer to Chapter 7, Managing Object Set Security, on page 87 for more detail about the use of object sets and security on object sets.

Summary of Checks When Accessing Objects

Sequence of Checks

The following sequence of checks takes place before your user ID can access an object:

1. The clearance level of the user ID is evaluated.
2. The ownership of the object is evaluated.
3. Access permissions are checked.

Clearance Level of the User ID

If your user ID clearance level is …	Access is …
7 (that is, a system administrator)	Allowed
Greater than the classification level of the object	Allowed
Less than the classification level of the object	Disallowed
Equal to the classification of the object	Checked

Ownership of the Object

If your user ID is …	Access is …
The owner user ID of the object	Allowed
The security administrator user ID that manages the object owner's user ID	Allowed
Not the owner user ID of the object	Checked

Access Permissions

If your user ID is …	and	Permitted accesses are taken from
Explicitly specified in the permission list for the object	n/a	Your user ID.
Not explicitly specified in the permission list for the object	In a current group that has access specified	The security group from which your user ID is operating.
Not explicitly specified in the permission list for the object	Not in a current group that has access specified	The security group ALL.

Chapter 3 Managing User Profiles

This chapter describes how to manage user profiles.

Topics

Creating and Accessing a User Profile

User profiles are used to store login defaults for a particular user ID. A user profile contains information about the operational parameters, the TIBCO Object Service Broker password, the login parameters, the printer options, and the application development parameters for a specific user ID.

Specialized user profiles are created and maintained for security administrators. Refer to Chapter 4, Managing Security Administrator Profiles, on page 43 for more information.

Who Can Access a User Profile?

A user profile can be accessed by the following people:

- The user associated— via login—with the user ID
- The security administrator who owns the subject user ID
- A system administrator

Who Can Create, Modify, and Delete a User Profile?

Type of User	Create	Modify	Delete
System administrator.	Y	Y	Y
Security administrator—without create user capability, or the non-owner of a subject user ID.	N	N	N
Security administrator—with create user capability.	Y	N	N
Security administrator—owner of a subject user ID with modify user capability.	N	Y	N
Security administrator—owner of a subject user ID with delete user capability.	N	N	Y
User associated with the login user ID.	N	Y [a]	N

a. The user associated with the user ID can modify only some of the values contained in the user profile, as described in the following sections.

Security failure message produced when accessing the underlying metadata tables used for the User Profile will indicate an access mode of VIEW_USER, MODIFY_USER, or MOD_USER, depending on the above criteria.

Creating Multiple User Profiles

If you have *create users* authorization you can create multiple user IDs based on a model user profile. For more information, refer to Creating Multiple User IDs on page 40.

Accessing a Profile

You can access a user profile in two ways:

- Directly from the standard developer's workbench, using the User Profile facility
- From the Security Management main menu

  You must use this method to create a user profile.

From the Developer's Workbench

To display your own user profile, from the developer's workbench position your cursor beside the option UP User Profile and press Enter, or execute the command directly as follows:

```
COMMAND ==> UP
```

From the Security Management Main Menu

After invoking the Security Manager facility, described in Accessing the Security Manager on page 5, you are presented with the Security Management main menu. Using this menu:

- You can access your own profile.

  To access your user profile, place the cursor beside the **User Profile** field and press Enter
- If you are security or system administrator, you can create or access the profiles of the security administrator's subjects.

  To create, view, or update the profile of another user, type the user ID in the **User Profile** field and press Enter. If you are creating a new user profile, use PF1 to display valid values for this field.

Sample User Profile

With either access method, the MANAGE USERID screen, illustrated below, appears:

```
Command ===>
--------------------------------------------------------------------------------
                   MANAGE USERID: HRSTAFF1           0     ( Clearance = 1 )
--------------------------------------------------------------------------------
Full Name: Joan Perkins                        Phone: x53753       Timezone: 0
CURRENT GROUP:                          SecAdmin: ADMN01

        Change Password                           Logon Parameters
------------------------------          ----------------------------------
Password:                               Logon Restricted from 1800 to 0800
Verify Password:                        Session Menu: STANDARD
                                        Security Group:
Ext Security Mixed-case Password: N     Library:
                                        Startup Rule: HRMENU6
       Print Parameters                 Action:     Browse:     Search:
------------------------------
Destination:                                    Application Parameters
Form:          Class: Y                 ---------------------------------------
FCB:           UCS:                     Character Set:
Number of Copies:   1                   Borrower: N      Default Unit: HRSTAFF1
External Writer:                        TDS Segment: 1
File:
PFKEYS: 1=HELP ENTER=CHANGE GROUP 5=MEMBERSHIPS 3=SAVE 12=CANCEL 22=DELETE
```

What Types of User Options Can be Set?

Using the Security Manager, you can set:

- Operational options
- Login options
- Startup rule options
- Print options
- Application development options

These are described in the following sections.

Setting Operational Options

Use the operational options to determine the security clearance, name and phone number, the time to use for local time, the security group to start a session from, the name of the security administrator, and the password for the user ID. The fields that control these options are described below.

MANAGE USERID and Clearance Fields

Values for the **MANAGE USERID** and **Clearance** fields are entered by default. If you have adequate security authorization, you can modify the value for security clearance. To display valid values for each field, use PF1.

MANAGE USERID	Displays the user ID being defined Required: Yes User Modifiable: No
Clearance	Displays the clearance level of the user ID. By default, any object created by this user ID takes on this clearance-level number as its classification level. A user ID is permitted access to any object only if its clearance level is equal to or greater than the object's classification level. If this clearance check fails, the object is not accessible, superseding any permission granted to the user ID. Assign level-7 clearance to create a system administrator. Required: Yes User Modifiable: No

Full Name, Phone, Timezone, CURRENT GROUP, and SecAdmin Fields

The information in the **SecAdmin** field is entered by default and cannot be modified. Refer to Chapter 4, Managing Security Administrator Profiles, on page 43 for information about security administrators. To display valid values for each field, use PF1.

Full Name	Specifies the name to be associated with the user ID. The Display Users facility uses the value in this field to assist in system administration. User Modifiable: No Required: No

Phone	Specifies the phone number to be used for this user ID. The Display Users facility uses the value in this field to assist in system administration. User Modifiable: Yes Required: No
Timezone	Specifies the time zone to be used for displaying time on the workbench menu User Modifiable: Yes Required: Yes
CURRENT GROUP	Specifies the current group in effect for this session. At the start of a session, the current group for a user ID defaults to the group specified in the **Security Group** field. The current group affects the permissions that are checked, granting or denying the user ID access to objects, and setting the default permissions assigned to new objects created by that user ID. Values in this field can be changed only by the user ID logged in for this session (that is, the user ID entered in the **MANAGE USERID** field). User Modifiable: Yes Required: No
SecAdmin	Displays the user ID of the security administrator assigned to this user ID. Refer to Obtaining Security Administrator Subjects on page 47 for information on assigning subjects to security administrators. User Modifiable: No Required: Yes

Password, Verify Password, and Mixed-case Password Fields

To display valid values for each field, use PF1.

Password and Verify Password	Use these fields to change a password. First enter the new password in the **Password** field and then enter the same password in the **Verify Password** field. When you use PF3 to save, TIBCO Object Service Broker verifies that the two entries are identical. To protect the privacy of the password, the contents of these fields are not shown. A null password is allowed on z/OS: The user ID can log in only if the TIBCO Object Service Broker user ID is the same as the external environment user ID (for example, a TSO user ID) and the SECURITY Execution Environment parameter is set to MIXED or EXTERNAL. User Modifiable: Yes Required: No
Ext Security Mixed-case Password	If you set the value of this field to Y, external security is used and mixed-cased password are accepted according to the way your security software accepts them. External security is used regardless of the setting of the Execution Environment SECURITY parameter. The following characters are not allowed regardless of whether or not your security software accepts them: ( ) + ` = { } [ ] \ " ; ' < > , / space. The default value is N. User Modifiable: Yes Required: No

If you have set a password using characters available with a keyboard, for example an accented letter such as é, your password may not be accepted if entered from another keyboard using alt key numeric codes, for example Alt+0233.

See Also

Case Sensitivity of Passwords on page 13 for information about whether your password must be uppercase for use on different platforms.

TIBCO Object Service Broker Managing Deployment about the Display Users facility.

TIBCO Object Service Broker Parameters about Execution Environment parameters.

Setting Login Options

Use the login options settings to determine login access times, the type of menu to appear, the default current security group, the rules library to use at login, the rule to use at session startup and the operational characteristics of the rule.

Login Restricted …, Session Menu, Security Group, and Library Fields

To display valid values for each field, use PF1.

Login Restricted from___to___	Specifies the time period when the user ID is *not* allowed to log in to TIBCO Object Service Broker. The format of this value is *hhmm* (for example, 2300 to 0530). User Modifiable: No Required: No
Session Menu	Specifies the session menu or workbench for the user ID User Modifiable: Yes Required: No
Security Group	Specifies the name that the **CURRENT GROUP** field defaults to at login time. Since this field is used only at login time, any changes to it affect the current group at the next login. Refer to Full Name, Phone, Timezone, CURRENT GROUP, and SecAdmin Fields on page 29 for more information about current groups. User Modifiable: Yes Required: No
Library	Specifies the default local rules library. If this field is null, the home rules library for the user ID is used. User Modifiable: Yes Required: No

Startup Rule, Action, Browse, and Search Fields

The information entered into the **Startup Rule**, **Action**, **Browse**, and **Search** fields determines the rule that is invoked when the user ID logs in to TIBCO Object Service Broker along with the characteristics of that rule. To display valid values for each field, use PF1.

Startup Rule	Specifies the rule to be invoked (based on the value in the **Action** field) immediately after login. When the startup rule terminates, the session menu (if specified) appears if the **Action** field is set to E or C. User Modifiable: Yes Required: No
Action	Specifies how the startup rule is invoked. It can contain one of the following: **C** – The startup rule is called. This enables the session manager to regain control of the session when the startup rule ends error-free. Otherwise, the session manager, not knowing how to handle the error exceptions from the startup rule, terminates with the session. The called rule must be in the system library. **E** – The startup rule is executed. This is the recommended value. Control is always transferred to the session menu, even if the startup rule finishes in error. As usual, executing a rule causes it to run at another transaction level. **T** – The startup rule is transfercalled. This terminates the session manager when the startup rule begins and the session then ends when the startup rule finishes. User Modifiable: Yes Required: No
Browse	Specifies whether (Y) or not (N) the startup rule is run in browse mode. It is specified only for action values of E or T. User Modifiable: Yes Required: No

Search	Specifies the search path for the startup rule. It is specified only if the action value is T or E, and can contain one of the following: **S** – System library: Only the system library is searched. **I** – Installation library: The search order is installation library first, and then system library. **L** – Local library: The search order is local library, installation library, and then system library. Required: No User Modifiable: Yes

See Also *TIBCO Object Service Broker Programming in Rules* about rules libraries.

Setting Print Options

Use the print option settings for printing control in the z/OS environment. If you are running on other platforms, most of these options while valid are ignored or overridden by values provided by other mechanisms. Refer to *TIBCO Object Service Broker Parameters* for information about setting print options on other platforms.

Destination, Number of Copies, File Fields

The following values must be known to the system and be available for use by the user ID. To display valid values for each field, use PF1.

Number of Copies	Specifies the default number of output copies desired User Modifiable: Yes Required: Yes
Number of Copies	Specifies the default number of output copies desired User Modifiable: Yes Required: Yes
File	Specifies the name of a file to which you want printed output directed. If specified, output is directed to this file instead of to the printer named in the **Destination** field. User Modifiable: Yes Required: No

Form, Class, FCB, UCS, and External Writer Fields

The following print parameters must be known to the system and be available for use by the user ID. To display valid values for each field, use PF1.

The following parameters are valid for z/OS only:

Form	Specifies that output data should be printed or punched on a special output form User Modifiable: Yes Required: No

Class	Specifies the class of your printed output Required: Yes User Modifiable: Yes
FCB	Specifies the type of output format User Modifiable: Yes Required: No
UCS	Specifies the default character set that should be used for printing the output data set User Modifiable: Yes Required: No
External Writer	The following parameter is valid for z/OS and Solaris only: Specifies a program that directs output to non-standard devices. If you specify a node name in the **Destination** field, leave this field blank. User Modifiable: Yes Required: No

Setting Application Development Options

Use the application development options settings to determine the national character set, whether or not objects created or modified by the user have promotion rights, the default working unit, and the storage location to be used for data.

Character Set, Borrower, Default Unit Fields

To display valid values for each field, use PF1.

Character Set	Specifies the national character set to use. The specification of a character set does *not* interchange commas and periods in numerics. The DECIMALSEPARATOR session parameter sets the decimal separator symbol. If NLS is enabled, the value for character set is ignored. User Modifiable: Yes Required: No
Borrower	Specifies whether or not the user ID can obtain promotion rights on objects that are created by it User Modifiable: No Required: Yes
Default Unit	Specifies the default unit to be assigned to objects when they are created. This default can be overridden when defining an object. User Modifiable: Yes Required: Yes

TDS Segment Field

To display valid values for this field, use PF1.

TDS Segment	Specifies the default segment, identified by its number, to store all new TDS data created by this user ID User Modifiable: No Required: Yes

See Also *TIBCO Object Service Broker National Language Support* about use of national character sets.

TIBCO Object Service Broker Managing Deployment about promotion rights.

TIBCO Object Service Broker Parameters about the DECIMALSEPARATOR parameter.

Saving and Deleting a User Profile

Who Can Save Changes to a User ID?

Both the user logged in under a given user ID and the owning security and system administrators can save modifications made to the user ID. If you are the owning security administrator, you must also have modify users authorization in your security administrator profile.

Only the owning security and system administrators can delete a user ID. If you are the owning security administrator, you must also have delete users authorization in your security administrator profile.

Saving a User Profile

Except for the **CURRENT GROUP** field, changes you make on the user profile screen take effect only after you press PF3 to save. A change to the current group takes place when you press Enter. It remains in effect whether you press PF3 to save or PF12 to cancel from this screen. To cancel any other changes, press PF12.

Deleting a User Profile

Before you delete a user ID, ensure that ownership for all of the user ID's objects are transferred to other user IDs as appropriate. When a user ID is deleted, any objects still owned by the user ID are transferred to the security administrator responsible for the user ID. If the user ID being deleted *is* a security administrator, and then ownership is transferred to the system administrator.

To delete a user ID, and all references to it in TIBCO Object Service Broker, access the user profile and press PF22. When you are prompted to confirm the delete request, press PF22 again.

The system administrator user ID, SYSADMIN, *cannot* be deleted.

Creating Multiple User IDs

To add a number of user IDs, you can use the CREATEUSERS tool to add them all at once, instead of adding them one by one. You must be a system administrator or a security administrator with create users authorization to use this tool.

Prerequisites

Before you execute the tool, you must define the following:

- A table that contains the user IDs (represented by the argument *input_table*)
- A model user profile on which the new user IDs are based (represented by the argument *modeluserid)*

Creating the Input Table

Create the input table using the Table Definer. The table must have a field named HURON_USERID (I C 8), to contain the listing of user IDs.

If the user IDs already exist in another security package such as RACF, you can extract the user IDs from the package and put them into the **HURON_USERID** field of the input table. If you use this method, you can define your input table as an import (IMP) table.

Creating the Model User Profile

Create the model user profile using the MANAGE USERID screen. The login, print, and other parameters specified for the model user profile are inherited by new user IDs. Individual users can further customize their profiles when they log in to TIBCO Object Service Broker.

Use of CREATEUSERS

To run CREATEUSERS, execute the tool as follows:

```
EX Execute Rule ==> CREATEUSERS <Enter>
```

A screen appears, prompting you for:

- The name of the *input_table*
- The *modeluserid*

After pressing Enter you are returned to the workbench and a message prompting you to press PF2 to see the results appears. If a user ID already exists, a message indicating this appears in the message log.

See Also *TIBCO Object Service Broker Shareable Tools* about the tools.

TIBCO Object Service Broker Managing Data about defining tables.

TIBCO Service Gateway for Files Installing and Operating about defining and using import tables.

Chapter 4 Managing Security Administrator Profiles

This chapter describes how to manage administrator profiles.

Topics

Creating and Accessing a Security Administrator Profile

Purpose of a Security Administrator Profile

TIBCO Object Service Broker distinguishes a security administrator from other regular users by the presence of a security administrator profile. Each security administrator has an individual profile. The profile lists the subjects of the security administrator and specifies whether the security administrator can add, modify, or delete user IDs.

Who Can Create and Access Security Administrator Profiles?

You must be a system administrator to create, modify, or delete a security administrator profile. The following can access any security administrator profile:

- Security administrators
- System administrators

Accessing a Profile

To create or view a profile, place the cursor beside the **SecAdmin Profile** field from the Security Management main menu and do one of the following:

- Type in the user ID and press Enter.

  The user ID must already have been defined using the User Profile facility.
- Press PF2 to display a list for selection. Type an **S** in the line command field next to the desired security administrator user ID and press PF3.

Using either access method causes the Security Administrator Profile screen to appear. From this screen you can specify or view:

- Security administrator privileges
- A list of subjects (user IDs administered by this security administrator)

Sample Security Administrator Screen

```
Command ===>
-------------------------------------------------------------------------------
              MANAGE SECURITY ADMINISTRATOR: PROJMGR8
-------------------------------------------------------------------------------
 Full Name: Richard I. Sebastian

 Is SecAdmin allowed to:

     Create Users?(Y/N): Y  Update Users?(Y/N): Y  Delete Users?(Y/N): N


 The following Users are administered by this SecAdmin:

       UserID             User Full Name
     --------  ----------------------------------------
   _ ARCT      R. Charles Tucker
   _ BUC2G     Bruce Conway (ASE)
   _ D23LGF    Lois Fairbank
   _ MTHCORP   Marge Heath
   _ ZOM       Zoltan Mahabir
   _ USR08     Ursula S. Romer

PFKEYS: 1=HELP 3=SAVE 12=CANCEL 4=DISCLAIM 6=ADD 22=DELETE
```

Specifying Security Administrator Privileges

As a system administrator, you can give a security administrator privileges to:

- Create user profiles—the security administrator then becomes the owner of the user IDs
- Update the user profile of a subject
- Delete the user profile of a subject

These privileges are mutually exclusive of each other.

Who Can Specify Security Administrator Privileges

Only a system administrator can specify security administrator privileges. Security administrators cannot make changes to the privileges assigned to them.

Adding Privileges

By default new security administrators are given no privileges. To add privileges, change the default value of N to Y in the appropriate fields of the Manage Security Administrator screen, for example:

```
Is SecAdmin allowed to:

  Create Users?(Y/N): N           Update Users?(Y/N): Y      Delete Users?(Y/N): N
```

In this example, the security administrator cannot create or delete users but can modify the profiles of subject user IDs.

Changing Privileges

To change the privileges assigned to a security administrator, change the Y/N value for appropriate privileges as required. The new values take effect after you exit using PF3 Security Administrator Subjects

Obtaining Security Administrator Subjects

A new security administrator has no subjects. Security administrators obtain subjects when:

- Creating new user IDs
- Subjects are referred by another security administrator
- The system administrator adds subjects to the security administrator

Restriction

You cannot add a subject to a security administrator's profile if the clearance level of the subject is greater than the administrator's.

Adding Subjects to Your Own Profile

If you are a security administrator, when you create a user ID it is automatically added as a subject to your security administrator profile. For example:

```
The following Users are administered by this SecAdmin:

   UserID          User Full Name
  --------   ------------------------------------------
_ ARCT       R. Charles Tucker
```

If you want to claim an existing user as your own subject but you are not a system administrator, the current security administrator of the subject or a system administrator must make the referral by disclaiming and transferring the subject to you. Refer to Adding Subjects to Another Security Administrator's Profile below for information about this procedure.

Adding Subjects to Another Security Administrator's Profile

The following people can add subjects to someone else's security administrator profile:

- If you are a system administrator, you can move subjects from one security administrator profile to another security administrator profile.
- If you are the owner of subjects, you can disclaim subjects you own and add them to another security administrator's profile. Refer to Removing Subjects From the Subjects List on page 49.

Adding Subjects

From the Manage Security Administrator screen (refer to Accessing a Profile on page 27), complete the following steps:

1. Press PF6 to display a list of user IDs for selection.
2. Type **S** in the line command field of the subject or subjects you are adding.
3. Press PF3.

   When you press PF3, the Manage Security Administrator screen reappears with the newly selected subjects.

Removing Subjects From the Subjects List

You can remove subjects from your security administrator profile by disclaiming them and referring them to another security administrator. Disclaiming a subject removes the user ID from the list of subjects in your profile.

System administrators can also remove subjects by disclaiming them first or by simply adding a user ID to a security administrator profile. This automatically removes the user ID from the original owner's subjects list.

Disclaiming and Transferring Subjects

To disclaim and transfer subjects from your profile to another, invoke your security administrator profile by entering your user ID in the **SecAdmin Profile** field on the Security Management main menu. Your subjects are listed on the bottom half of the screen. From this screen, complete the following steps for each subject you are disclaiming:

1. Position your cursor beside the user ID of the subject you are disclaiming.
2. Press PF4 to display the screen used to select a new security administrator.
3. Type **s** beside the name of the new security administrator.
4. Press PF3.

   When you press PF3 the Manage Security Administrator screen reappears with a message similar to:

   ```
   User "D23L" DISCLAIMED; on SAVE, new SecAdmin will be "HR0002"
   ```

You can save the profile now using PF3 or disclaim other subjects in your list.

Deleting a Security Administrator Profile

Prerequisite for Deleting a Profile

Deleting the security administrator profile removes all security administration privileges from that user ID. There are two prerequisites for deleting the profile:

- There can be no existing subjects for the security administrator.

  Refer to Removing Subjects From the Subjects List on page 49.

- Deletion can be done only by a system administrator.

Deleting the Profile

When the subjects of the security administrator have been transferred, complete the following steps to delete the profile:

1. Invoke the security administrator's profile.
2. Press PF22 to delete the profile.
3. When you are prompted to confirm your delete request, press PF22 again.

Deleting the security administrator profile removes only administrator privileges from the user ID. The user ID remains in the TIBCO Object Service Broker system and is still usable.

Chapter 5 Managing Security Groups

This chapter describes how to manage security groups.

Topics

Accessing Security Groups

A security group is a list of users who can be assigned permissions on a group basis. When users operate out of a specific group, they get the permissions assigned to the group. The use of security groups simplifies the process of managing a group of users with the same security needs.

A user can be a member of various security groups but can operate only out of one group—the current group—at one time. For more information about how to set your current group, refer to Full Name, Phone, Timezone, CURRENT GROUP, and SecAdmin Fields on page 29.

Who Can Create and Access Security Groups?

Both users and administrators can create new security groups as the need arises. A user who creates a security group becomes the owner of the group.

When a security group is defined, your security privileges determine if you can access information about a group:

- Users who are members of a security group can view that group's information, provided the group's owner gives them viewing rights.

  The Manage Security Group screen, illustrated in Sample Security Group Screen on page 53, lists the members of the group and indicates whether or not each member can view the group specification.

- The owner of a security group and the owner's security administrator can view and update membership and other information for that group.
- System administrators can view and update all security group information.

Security failure message produced when accessing the underlying metadata tables used for Security Groups will indicate an access mode of VIEW_GROUP, MODIFY_GROUP, or MOD_GROUP, depending on the above criteria.

Accessing the Security Group Screen

To display the Manage Security Group screen, place the cursor beside the **Security Group** field on the Security Management main menu and do one of the following:

- Type in the name of a security group and press Enter.

- Press PF2 to display a list for selection. Type an **S** in the line command field next to the desired group name and press PF3. When the selected name appears in the Security Group field, press Enter.

Using either access method causes the Manage Security Group screen to appear. From this screen you can:

- Update the member information
- View the members of the groups included in this security group
- View the object sets enabled for this group

Sample Security Group Screen

```
Command ===>
 ------------------------------------------------------
                MANAGE SECURITY GROUP: PROJMGR8
 ------------------------------------------------------
Group Description:Information Center(HR Unit) Creator: PROJMGR8
           Name          User or Group            View
      USERID | GROUP      Description              Allowed
      --------------     ---------------------      -
      BGLC               Bertrand Gould             Y
      DED7               Debbie Dunning             Y
      JMH5               John Hallsworth            Y
      PROJMGR8           Richard I. Sebastien       Y
      TEMP12             Howie Langdon (co-op)      N
      TMF3B              Tom Ferguson               Y




PFKEYS:1=HELP 6=USERIDS 9=GROUPS 21=VIEW 4=OBJ 3=SAVE 12=CANCEL
```

Creating a Security Group

To create a new security group, place the cursor beside the **Security Group** field on the Security Management main menu. Type in the name of the new group. For valid values for the name, use PF1. After typing in a valid name, press Enter to display a screen similar to the one in Sample Security Group Screen on page 53.

Adding Users to a Group

You can add users to the group by:

- Typing in the user IDs that make up the new security group
- Selecting user IDs from a list of all user IDs
- Selecting user IDs from other security groups

A security group *cannot* include other security groups as members.

Selecting from a List of User IDs

From the Manage Security Group screen, complete the following steps to select from a list of user IDs:

1. Press PF6 to display a list of all user IDs.
2. Type **s** next to the user ID(s) to select user IDs from this list.
3. Press PF3 to transfer your selection and return to the Manage Security Group screen.

Selecting Users from Other Groups

From the Manage Security Group screen, complete the following steps to select user IDs from other groups:

1. Press PF9 to view a list of security groups.
2. Type **s** next to the group(s) that includes potential members for your new group.
3. Press PF3 to transfer your selection and return to the Manage Security Group screen.
4. Position your cursor on a group name added to your membership list and press PF21 to view the membership of that group.

The Security Manager displays only the membership of the selected group if you are allowed to view its membership. Refer to Viewing Security Group Information on page 56.

5. From the list of group members displayed, select individual user IDs to be added to your group.
6. After selecting user IDs, delete the group name from the list of members and press PF3 to save this security group specification.

To cancel the security group specification and return to the Security Management main menu, press PF12.

Updating a Security Group

If you are viewing an existing security group that you own, you can also add, change, and remove members during the session. To remove users from a security group, erase the user ID from the membership list or overtype it with blanks and then press PF3 to save the changes.

To cancel any changes you made and return to the Security Management main menu, use PF12.

Viewing Security Group Information

Displaying Members of a Security Group

If you have permission to view the membership of an existing security group, complete the following steps to display the members:

1. Position your cursor beside the Security Group field on the Security Management main menu.
2. Type in the name of a known security group
3. Press Enter.

   This displays a screen similar to the one shown in Sample Security Group Screen on page 53.

Listing the Object Sets for a Security Group

To view a list of object sets enabled for a group, press PF4 from the Manage Security Group screen. The list displayed cannot be modified from this screen.

Object Sets for Group Screen

```
-----------------------------------------------------
       OBJECTSETS FOR GROUP: PROJMGR8
-----------------------------------------------------

     Objectset Name      Enabled
     ----------------       -
     DOCORDERAUDIT          Y









  PFKEYS: 1=HELP 3=SAVE 12=CANCEL
```

Viewing Your Security Group Affiliations

From the MANAGE USERID screen, you can view your membership information by pressing PF5. Membership lists appear for your information only; if you require changes, contact your security administrator.

From the Manage Memberships screen you can:

- View the security groups to which you have memberships
- Determine whether you can view the membership list of the group
- View the names of the object sets to which you have access
- Determine whether or not an object set is security-enabled

Refer to Chapter 7, Managing Object Set Security, on page 87 for more information about object set security.

MANAGE MEMBERSHIPS FOR USERID Screen

```
------------------------------------------------------------
             MANAGE MEMBERSHIP FOR USERID: HRSTAFF1
------------------------------------------------------------

     Member of Groups    View     ObjectSets      Enabled
     ----------------     -      ---------------     -
     NRCLASS4             N      STAFFREPORTS12      Y
     PAYROLLINQ           Y      EMPLOYEEARCHIVE     Y
     EMPDISCOUNTREAD      Y      SALARYANALYSIS      Y
     ADHOCDEVTEAM         Y


  PFKEYS: 1=HELP 6=OBJECTSETS 9=GROUPS 3=SAVE 12=CANCEL
```

Chapter 6 Managing Permissions to Objects

This chapter describes how to manage the permissions to objects.

Topics

Ownership Privileges and Permissions

What are Ownership Privileges?

When you create an object you become its owner. Ownership of an object means that you have the following privileges:

- Unlimited access to the object
- The ability to grant other user IDs and groups permissions to the object
- The ability to delete the object
- The ability to relinquish your ownership of the object

Who Shares Owner Privileges?

Ownership privileges are shared by:

- You, as the object owner
- Your security administrator
- A system administrator

What are Permissions?

Permissions are access rights that you as the owner of an object grant to others. As the owner, you can specify the following types of permissions:

- Control
- Explicit
- Default

Control permission determines who manages access rights to an object and explicit and default permissions determine how access rights are assigned.

Refer to Setting up Explicit Permissions to Objects on page 62 for a description of explicit permissions and how to assign them. Refer to Setting Up Default Permissions on page 78 for a description of default permissions and how to assign them.

Control Permission

Control permission allows a user ID to manage access rights to an object. If you are the owner of the object, you automatically have control permission. You can also allow other user IDs to manage permissions to the object by granting them control permission.

The following conditions apply to granting control permission:

- You must have ownership privileges on an object to grant control permission to others.
- Other user IDs with control permission can assign and delete permissions to the object but they cannot grant control permission to other user IDs.
- You *cannot* assign control permission to a security group.

Object Permissions and Enabled Object Sets

Permissions cannot be changed for an object that is part of an object set that is currently enabled. If an object is part of the definition of an object set that has its status set to enabled, you see a message on the bottom of your screen stating:

```
WARNING: object is part of ENABLED object set; can't SAVE
```

In this case, use PF4 to determine the names of the enabled object sets that use the object. Use PF12 to exit from the permissions specification.

For information about enabled object sets refer to Enabling an Object Set on page 96 and for information about changing the permissions, refer to Updating Object Permissions When an Object Set is Enabled on page 100.

Setting up Explicit Permissions to Objects

What are Explicit Permissions?

Explicit permissions are permissions granted on an individual object basis and, in the case of a parameterized table, on a parameter value basis. They are assigned to a user ID or group of user IDs through the Manage Permissions To Objects interface.

Granting Explicit Permissions for Objects in an Object Set

Explicit permissions for objects that make up an object set are granted using the Manage Permissions to Object Sets interface. Refer to Managing Permissions to Objects in an Object Set on page 89 for more information.

Tasks Required to Explicitly Manage Permissions

When the main menu appears, complete the following tasks to explicitly manage permissions to an object:

	Task	Refer to
A	Identify the object.	63
B	If the object is a parameterized table, identify the parameter values for which you want to assign permissions.	66
C	Identify the classification level of the object.	69
D	If the object is a table, indicate whether to log accesses to it.	70
E	Identify the access modes for the object.	72

Task A: Identify an Object

Because explicit permissions are granted at an individual object level, you must identify the object that you require. Also, because object names are unique only within an object type, you must also identify the object type of the object on which you are granting permissions.

Methods Available to Identify an Object

There are two ways to identify an object on the menu:

- If you know the object name, you can type it in directly.
- If you do not know the object name, you can display a list of objects and select from the list.

Both methods are described in the following sections.

Typing in the Object Name

If you know the name and type of object that you are specifying permissions for, complete the following steps:

1. Position the cursor beside the Object field and type in the object name.

   The object you name must exist and you must either own it or have control permission to it. If the object is a table, the table must be defined, although it can be empty of data.

2. Position the cursor beside the Type field and specify the object type, using one of the following values:

TBL	Table
SCR	Screen
RPT	Report
LIB	Library

3. Press Enter.

   This displays a screen similar to the example shown in Specify Protection Screen on page 65.

Selecting From a List

To display a list of objects and select from the list, complete the following steps:

1. Position your cursor on the Type field and press the Options PF key PF2 to display a list of object types.
2. Select an object type by typing an **S** on the line command field beside the object type.
3. Press PF3.

   This transfers your selected object type and return you to the Security Management main menu.
4. Position the cursor on the Object field, in the Security Management main menu and press PF2.

   This displays a list of all objects of the selected type.
5. Select an object by typing an **S** in the line command area beside the object name.
6. Press PF3.

   This transfers your selected object's name and returns you to the main menu.
7. When the main menu reappears with the selected object name in the Object field, press Enter again.

   This displays a screen similar to the one shown in Specify Protection Screen below.

Specify Protection Screen

Use the Specify Protection screen, shown below, to specify protection for a table object type. Each object type (library, report, screen, and table) has its own Specify Protection screen. The layout of this screen is similar for each object type, except that you can also specify logging of accesses to table objects. Refer to Task D: Logging Accesses to a Table on page 70 for more information.

```
--------------------------------------------------------------------------------
                  SPECIFY PROTECTION FOR TABLE:  DEPARTMENTS
--------------------------------------------------------------------------------

Owned by:USR50     Smith, Joan                              Classification:  1
                                                              Log Accesses:  N

      Name                          Access Modes Allowed
(USERID | GROUP)   READ INS REPL DEL DEF_VIEW DEF_PRM MOD_DFN VIEW_DEFN CTRL
----------------    -    -    -    -     -        -       -        -      -
USR10                Y    N    N    N     N        N       N        Y      N




PFKEYS: 1=HELP 4=OBJECTSETS 9=GROUPS 21=VIEW 6=USERIDS 3=SAVE 12=CANCEL
```

Task B: Identify Table Instances

If your table is parameterized, you can specify permissions for individual instances or all instances of the table.

Permissions that you grant for a specific parameter value *override* permissions that you grant at the table level for all instances.

Sample Specify Protection for Table Screen

A screen similar to the one below appears when you select to specify permissions for a parameterized table from the Security Management main menu. In this sample screen, COUNTRY and DISTRICT are parameters belonging to the SERVICEDEPOTS table.

```
--------------------------------------------------------------------------------
                SPECIFY INSTANCE OF TABLE:  SERVICEDEPOTS
--------------------------------------------------------------------------------

Table Parameters:             COUNTRY =
                             DISTRICT =
( ALL DATA: Y )

     < specify all parameter values or set ALL DATA = Y for whole table >

PFKEYS: 1=HELP ENTER=CONTINUE 12=CANCEL
```

Prerequisite Step

To proceed from this screen, decide which of the following tasks you want to perform:

- Manage non-data access permissions
- Manage data access permissions on the entire table (all its instances)

- Manage data access permissions, that is, READ, INSERT, REPLACE, or DELETE, on specific table instances

These steps are described in the following sections.

Managing Non-data Access Permissions

The following permissions, which apply to the table itself and not to any specific instances, are not pertinent to data access:

- View a definition (VIEW_DEFN)
- Define calculation or subviews (DEF_VIEW)
- Define a parameter value table (DEF_PRM)
- Modify a definition (MOD_DFN)
- Control the object's permissions (CTRL)

Required Steps

To manage these permissions, complete the following steps:

1. Set the ALL DATA field to Y.
2. Press Enter.

   The screen for non-parameterized tables appears. Refer to Specify Protection Screen on page 65 for an illustration of this screen.
3. Proceed in the same way as with non-parameterized tables.

Data access permissions that you change here apply to the entire table, that is, without regard for its parameterization.

Managing Data Access Permissions to an Entire Table

To manage data access permissions for all instances of a table, do the following steps:

1. Set the ALL DATA field to Y.
2. Press Enter.

   The screen for non-parameterized tables appears. Refer to Specify Protection Screen on page 65 for an example of this screen.
3. Proceed in the same way as with non-parameterized tables.

Data access permissions that you change here apply to the entire table, that is, without regard for its parameterization.

Managing Data Access Permissions to a Specific Table Instance

To manage data access permissions to a specific table instance, do the following steps:

1. Set the ALL DATA field to N.
2. Identify the table instance you want to manage by supplying the specific parameter values for the instance.

   The table instance can be empty of data but you must enter values that are valid for the defined parameters. The screen shown below has a sample entry for each parameter.
3. Press Enter.

Screen to Specify Protection for an Instance

```
--------------------------------------------------------------------------------
                  SPECIFY PROTECTION FOR TABLE:  SERVICEDEPOTS
--------------------------------------------------------------------------------

Owned by: HC324F     Conrad Powers                        Classification:  1
                                                            Log Accesses:  N
Table Parameters:             COUNTRY = AUSTRALIA
                             DISTRICT = NSW2
( ALL DATA: N )


          Name                  Access Modes Allowed TABLE
       (USERID | GROUP)           READ INS REPL DEL
       ----------------            -    -   -    -




PFKEYS: 1=HELP 4=OBJECTSETS 9=GROUPS 21=VIEW 6=USERIDS 3=SAVE 12=CANCEL
```

On this screen you can add, change, and delete permissions, but only for the four data access permissions READ, INSERT, REPLACE, and DELETE. These permissions apply *only* to the instance whose parameter values are shown in the **Table Parameters** field.

Users or groups accessing a table also need, at least, permission to view the definition of the table. You must give VIEW_DEFN permission separately to the same users. This procedure is explained in Managing Non-data Access Permissions on page 67.

Task C: Modify the Classification Level

Use this optional task to modify the default value provided for the object's classification level. The classification level of a new object defaults to your security clearance level. You must complete this task if you require a different classification level for the object. Refer to Clearance Checks When Accessing Objects on page 20 for additional information about clearance and classification levels.

Modifying the Default Value

From the Specify Protection For Object Type screen, overtype the default value in the **Classification** field with a new value. Valid values for this field are 1 or 7.

Making an Object Inaccessible to Its Owner

If an object's classification level is changed to a value that is higher than the security clearance level of the user ID that owns it, the object becomes inaccessible to that user ID. This change takes effect the next time the user ID logs in or when bound security information is refreshed, whichever occurs first. Refer to Binding of Security Access Data on page 132 for more information about bound security information.

Task D: Logging Accesses to a Table

By default, user accesses are *not* logged for new table definitions (that is, **Log Accesses=N**). However, the owner of an object and the owner's security administrator can specify whether or not to log user accesses to non-dictionary (MetaStor) tables. All user accesses (READ and UPDATE) are then logged.

When the **Log Accesses** field is set to Y, commit levels could be reached on large tables. Set this field to Y with caution.

Overriding Values

In the following circumstances, supplied values for the **Log_Accesses** field are overridden:

- If the table is being promoted, the value for this field is determined by the Promotions facility.
- If logging is disabled, the value for this field is ignored. Refer to Chapter 8, Auditing Accesses, on page 103 for additional information about security logging.

Required Steps

To modify the value for logging user accesses, complete the following steps:

1. From within the Security Manager, access the Specify Protection for Table screen (shown in the illustration below) for the appropriate table.
2. To turn logging on, type Y in the Log Accesses field.

   To turn logging off, type N.

Specify Protection Screen

```
--------------------------------------------------------------------------------
                  SPECIFY PROTECTION FOR TABLE:  DEPARTMENTS
--------------------------------------------------------------------------------

Owned by:USR50     Smith, Joan                              Classification:  1
                                                              Log Accesses:  N

      Name                          Access Modes Allowed
(USERID | GROUP)   READ INS REPL DEL DEF_VIEW DEF_PRM MOD_DFN VIEW_DEFN CTRL
----------------    -    -    -    -     -        -       -        -      -
USR10                Y    N    N    N     N        N       N        Y      N









PFKEYS: 1=HELP 4=OBJECTSETS 9=GROUPS 21=VIEW 6=USERIDS 3=SAVE 12=CANCEL
```

See Also *TIBCO Object Service Broker Managing Deployment* about promotions.

Task E: Set Accesses

Each object type has specific access modes that can be assigned to it. Use this task to assign the modes based on user IDs or groups to which you are allowed to give permissions.

Steps Required

To grant accesses, complete the following steps:

1. From within the Security Manager, access the Specify Protection for object name screen.

   The screen used for tables is shown in the illustration below.

2. To specify the access for each user ID or group, type Y (access allowed) or N (access not allowed) in the appropriate Access field.

Specify Protection for Table Screen

```
--------------------------------------------------------------------------------
                 SPECIFY PROTECTION FOR TABLE:  DEPARTMENTS
--------------------------------------------------------------------------------

Owned by:USR50     Smith, Joan                          Classification:  1
                                                          Log Accesses:  N

      Name                          Access Modes Allowed
(USERID | GROUP)   READ INS REPL DEL DEF_VIEW DEF_PRM MOD_DFN VIEW_DEFN CTRL
----------------    -    -    -    -     -        -       -        -      -
USR10                Y    N    N    N     N        N       N        Y      N




PFKEYS: 1=HELP 4=OBJECTSETS 9=GROUPS 21=VIEW 6=USERIDS 3=SAVE 12=CANCEL
```

Different Accesses for Each Object Type

Different object types have different accesses allowed. A table, for example, has accesses that apply to its data contents in addition to its definition. Because the accesses of the various object types are different, each object type has its own screen to specify protection.

Valid Accesses Available

The following list describes the valid accesses that are available:

READ	View data in an object.
INSERT	Insert data into an object.
REPLACE	Modify the data in an object.
DELETE	Delete the data in an object.
DISPLAY	Display the screen using the DISPLAY statement.
PRINT	Process the report using the $SETRPTMEDIUM tool, the $RPTPRINT tool, or the PRINT rules language statement.
DEF_VIEW	Define calculation views (CLC table type) and subviews (SUB table type) on the table.
DEF_PRM	If the table is parameterized, define a table to hold the parameter values of the table (PRM table type).
MOD_DFN	Modify the definition of the object.
VIEW_DEFN	View the definition of the object.
CONTROL	Control the permissions to the object.

Valid Accesses by Object Type

The following table lists the valid accesses for you to specify for each object type. A "Y" indicates that the access can be specified for a given object type.

	VIEW_DEFN	READ	INS	REPL	DEL	DISPLAY	PRINT	DEF_VIEW	DEF_PRM	MOD_DFN	CTRL
Table	Y	Y	Y	Y	Y			Y	Y	Y	Y
Screen	Y					Y				Y	Y
Report	Y						Y			Y	Y
Library	Y	Y	Y	Y	Y					Y	Y

See Also *TIBCO Object Service Broker Shareable Tools* about tools.

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

Modifying Permissions

The permissions that you assign to your objects can be modified as the need arises. The following sections describe how you can add, change, and delete permissions to your objects.

Adding Permissions

To assign additional permissions to an object, you can add user IDs or group names to the object permissions list on the Specify Protection for … screens in any of the following ways:

- Enter the name of a user ID or group, press Enter, and then modify the appropriate Y or N values to match the accesses required.

  You cannot assign control permission to a group.

- Press the USERIDS PF key PF6 to select users from a list of all available user IDs.
- Press the Groups PF key PF9 to select group names from a list. Position your cursor on the name of a group and press PF21 to view and select the users from a security group.

To save additions to the object permissions, press PF3. To cancel them, press PF12.

Changing Permissions

To change the access modes allowed for any user or group, change the Y or N values accordingly, and press PF3 to save your changes. To cancel your changes, press PF12.

Deleting Permissions

To delete permissions to an object, erase (or overtype with blanks) any user ID or group names that should be deleted from the permissions list and press PF3. To cancel the deletions, press PF12.

Transferring Ownership of Objects

You can transfer ownership of libraries, screens, reports, tables, security groups, and object sets from your user ID to another user ID. Each object retains its security clearance when it is transferred to a new owner.

Invoking the Transfer Ownership Screen

To transfer ownership, from the Security Management main menu position your cursor beside the **Transfer Ownership** field and specify one of the following:

- If you want to transfer ownerships from the user ID currently logged in to TIBCO Object Service Broker, leave the **User Profile** field blank.
- If you are a security administrator, you can enter the user ID of one of your subjects in the **User Profile** field.

When you press Enter, a screen similar to the one below appears.

Sample Transfer Ownership Screen

```
--------------------------------------------------------------------------------
   Transfer Ownership of Objects Owned by  USR40     to New Owner
--------------------------------------------------------------------------------
 Search for type=     & name like

         Tables                  Screens                  Reports
  ------------------       ------------------       ------------------
  _ DOCBODY                _ DOCMAINTENANCE         _ DOCPICKINGLIST
  _ DOCFOOTER              _ DOCORDERENTRY
  _ DOCHEADER
  _ DOCORDER
  _ DOCORDERITIEMS
  _ DOCPICKINGLIST
      Libraries                 Groups                Objectsets
  ------------------       ------------------       ------------------
                           _ APDEV                  _ DOCORDERENTRY
                           _ HRDBADM                _ DOCORDERPROCESS

 <Place an "S" beside the objects for which ownership will be transferred>


PFKEYS: 1=HELP 6=USERIDS 5=SEARCH 3=TRANSFER&EXIT 12=CANCEL
```

Required Steps

To transfer ownership, complete the following steps:

1. Specify the user ID you want to transfer the objects to in the New Owner field (top right corner).
2. If you do not know the user ID, press PF6 and select a user ID from the list displayed.
3. Select objects to transfer by typing an **s** to the left of the object.

   You can scroll each object list vertically pressing PF7 and PF8 to view the entire list.
4. If you own many objects of each type, consider using the SEARCH function on this screen.

   You can specify the object type (TBL, SCR, RPT, LIB, OBS, or GRP) in the **Search for the type=** field and a search mask, for example, PAY_*, in the **& name like** field. Press PF5 to position the cursor at the first occurrence that matches the search mask. Successive uses of PF5, if type and mask are not changed, move the cursor to the next match.
5. After selecting all objects to be transferred to the new owner, press PF3 to transfer ownership and return to the Security Management main menu.

   To return to the menu without changing ownership, press PF12

Exercise care when pressing PF3 and transferring ownerships. It is possible that you cannot undo transfers. Only the new owner or the owner's security administrator can relinquish the new ownership.

Setting Up Default Permissions

What are Default Permissions?

Default permissions are permissions granted automatically to a user ID or group when an object is created. When you want to control access to your newly created objects, setting up default permissions saves you from having to manage permissions to many individual objects.

To set up default permissions, define a default permissions list using the Specify Permissions Lists interface, available from the Security Management main menu. For more detail, refer to Creating and Accessing a Default Permissions List on page 81.

Default permissions can also be used to deny others access to objects.

Who Can Specify Default Permissions?

You can specify default permissions for objects created by:

- Your user ID
- Security groups for which you have responsibility

Your security administrator or a system administrator can also set up default permissions for your user ID.

As the object owner, you have the final word in allowing or denying others access to objects you own, subsequent to their creation when default permissions were applied. Changing default permissions does not affect permissions to your existing objects. You should change permissions to existing objects as described in Adding and Updating Default Permissions for Objects on page 83.

Default Permissions for Your User ID

You can specify default permissions for your user ID, so that you automatically assign others access to new objects you create. For example, you can set up a default permissions list that provides user SMITH with VIEW_DEFN, INSERT, and REPLACE access rights to new tables you create. When you have saved the default permissions list, any time you create a table, SMITH is automatically given VIEW_DEFN, INSERT, and REPLACE accesses to it.

Default Permissions for a Security Group

You can set up default permissions for a security group for which you have responsibility so that every time someone operating out of this group creates a table, all members listed in the default permissions list are given the specified accesses to it.

For example, the default permissions for the APDEV security group could specify that an object created by a member operating out of this group has the accesses shown in the table below assigned to it automatically, for the specified users and groups.

Group or User ID	Access Types
APDEV	READ, VIEW_DEFN, INSERT, REPLACE.
SYSTEMS	READ, VIEW_DEFN.
PROJLEAD	All accesses, including CONTROL.
Other groups and selected user IDs.	VIEW_DEFN and READ access.

When you specify a security group as your current group, new objects you create while you operate out of this group are also assigned the default permissions for that security group.

Assignment of Default Permissions at Object Creation

When you create an object, default permissions are assigned in the order below.

ALL	If default permissions for objects of the type created are specified for the virtual group ALL, these defaults are applied first.
Current Group	If default permissions for objects of the type created are specified for the current group, these defaults are applied next. These defaults could override the defaults applied at the previous level.
User ID	If default permissions for objects of the type created are specified to the user ID, these defaults are applied last. These defaults could override the defaults applied at the previous levels.

Effects of the Value for Current Group

The value of your current group affects how default permissions are assigned:

- If your current group is set to null, new objects you create during the session are assigned the default permissions specified for ALL and each user ID.
- If you specify a current group (for example, APDEV) for your session, new objects you create during the session are also assigned the default permissions specified for that Security Group.

Creating and Accessing a Default Permissions List

Who Can Create and Access a Default Permissions List?

Users, security administrators, and system administrators can create default permissions lists.

Except for the user group ALL, which all users can view, the following restrictions apply to accessing a default permissions list:

- The owning user ID can view and modify a default permissions list.
- User IDs or members of groups included in a default permissions list must have permission to access a defined permissions list.
- Security administrators can view and update default permission lists created by a subject.
- System administrators can view and update any default permissions list.

Invoking the Default Permissions Screen

To create or access a default permissions list, type one of the following values in the **Default Permissions** field on the Security Management main menu and do one of the following:

- Type in your user ID and press Enter.
- Type in the name of a security group for which you have responsibility and press Enter.
- Press PF2 to display a list for selection. Type an S in the line command field next to the desired name and press PF3. When the selected name appears in the **Default Permissions** field, press Enter.

Using any of these methods causes the Manage Default Permissions screen, shown below, to appear.

Manage Default Permissions Screen

In this sample screen, all members of the group APDEV automatically receive permissions specified to new tables, screens, reports, or libraries created by USR08.

```
------------------------------------------------------------------------------
    MANAGE DEFAULT PERMISSIONS for objects created by USR08
------------------------------------------------------------------------------


_ on create Table   _ on create Screen _ on create Report _ on create Library
----------------    ----------------   ----------------   ----------------
APDEV               APDEV              APDEV              APDEV
HRDBADM             HRDBADM            HRDBADM            HRDBADM
HR305               HR305              HR305
INFOCTR8            PROMADM6           PROMADM6
PROMADM6



PFKEYS: 1=HELP 4=TBLS 5=SCRS 6=RPTS 9=LIBS 3=SAVE 12=CANCEL
```

Data Displayed

The Manage Default Permission screen displays the creating user or group and displays a field to use to add group or user ID access to specific object types. If permissions are assigned, it also lists under each object type the other user IDs and groups who automatically receive permissions when the creating user ID or group creates an object of that type.

Adding and Updating Default Permissions for Objects

Specify Default Permissions

To specify the default permissions for each type of object, use these function keys:

PF4	Tables
PF5	Screens
PF6	Reports
PF9	Libraries

Allowed Changes

For each of the object types, you can:

- Change access modes
- Add new user IDs or group names
- Delete user IDs or group names
- Change user IDs or group default permissions

The following sections describe how to make these changes to your default permissions.

Sample Defaults Permission Screen for a Table Object

Use the Specify Default Permissions screen, shown below, to specify permissions for a table object type. Each object type (library, report, screen, and table) has its own Specify Default Permissions screen. The layout of this screen is similar for each object type.

```
--------------------------------------------------------------------------------
     Specify DEFAULT PERMISSIONS for TABLES created by  USR08
--------------------------------------------------------------------------------

        Name                          Access Modes Allowed TABLE
  (USERID | GROUP)    READ INS REPL DEL DEF_VIEW DEF_PRM MOD_DFN VIEW_DEFN CTRL
  ----------------     -   -   -    -     -        -       -        -      -
  APDEV                Y   N   N    N     N        N       N        Y      N
  HRDBADM              Y   Y   Y    Y     Y        Y       Y        Y      N
  HR305                Y   Y   Y    Y     Y        Y       N        Y      N
  INFOCTR8             Y   N   N    N     N        N       N        Y      N
  PROMADM6             Y   N   N    N     N        N       N        Y      N




PFKEYS: 1=HELP 6=USERIDS 9=GROUPS 21=VIEW 3=SAVE 12=CANCEL
```

For a description of each of the allowed access modes for tables, screens, reports, and libraries, refer to Valid Accesses by Object Type on page 74.

Changing Access Modes

To change the default access modes for any user or group, change the Y or N values accordingly and press PF3 to save your changes.

Adding New User IDs or Group Names

You can assign additional default permissions to user IDs or groups in any of the following ways:

- Enter the names of known user IDs and security groups.
- Press PF6 to select users from a list of all available user IDs.

- Press PF9 to select security group names from a list.
- Position your cursor on the name of a group and press PF21 to view and select the users from a security group.

To save any additions to the default permissions, press PF3. To cancel your changes, press PF12.

You can assign *control* permission only to user IDs, not groups.

Deleting a Member Name

To delete a user or security group from the Default Permission List, erase the member name or overtype it with blanks. Entries that have no member name specified are automatically removed when you use the next function key.

Changing a Member Name

To change a member name, type the new name over the existing name. The access modes remain as they were for the previous member until you explicitly change them.

Chapter 7 Managing Object Set Security

This chapter describes how to manage objects sets security.

Topics

Security for Object Sets

What is an Object Set?

A TIBCO Object Service Broker application is made up of one or more rules, tables, screens, and reports. Using the Object Set Definer workbench tool, you can define an object set. Using an object set, you can define and manage all the individual objects that make up your application.

You grant security on objects within an object set through the use of an object set permissions list. You can activate security from within the Object Set Definer to set up this list or set it up directly from the Security Manager. Refer to *TIBCO Object Service Broker Application Administion* for information about using the Object Set Definer.

Who Can Set Security Permissions to an Object Set?

The following people can set permissions to an object set:

- The user associated—via login—with the user ID used to create the object set

  This user ID is the owner of the object set.
- The security administrator who is the owner of the subject user ID
- A system administrator

What is an Object Set Permission List?

The list of security permissions associated with an object set is called an object set permissions list. As the owner of the objects in the object set, you must grant users and groups appropriate security accesses to each object in the set. Rather than maintain each object's permissions individually, you can set up a list of objects and their associated permissions and manage the object set as a whole from within the Security Management facility. Refer to Object Set Permissions on page 89 for more information.

Activating Object Set Permissions

Before the permissions specified in the permissions list can be applied, you must first activate them. This activation process, known as enabling an object set, can be done interactively or through a batch job. Refer to Enabling and Disabling Object Sets on page 96 for more information.

Object Set Permissions

Managing Permissions to Objects in an Object Set

Accessing the Manage Object Set Permissions Screen

To create an Object Set Permission List from the Security Management main menu, position your cursor beside the **ObjectSet Permissions** field and do one of the following:

- Type in the name of the object set.
- Press PF2 to display a list for selection. Type an S in the line command field next to the desired object set name and press PF3. When the selected name appears in the **ObjectSet Permissions** field, press Enter.

Using either method causes the Manage Object Set Permissions screen, illustrated below, to appear. If you are defining an object set on the Define Objectset screen, you can save your definition and transfer to this screen by pressing PF4.

Manage Object Set Permissions Screen

```
 Command ===>
------------------------------------------------------------------------------
               MANAGE OBJECTSET PERMISSIONS: DOCORDERENTRY
------------------------------------------------------------------------------
 Description: Document Order Entry + Production              Creator: PROJMGR8

 _     Tables          _    Screens         _    Reports         _    Libraries
 ----------------      ----------------     ----------------     ----------------
 DOCBODY               DOCMAINTENANCE       DOCPACKINGLIST
 DOCFOOTER             DOCORDERENTRY
 DOCHEADER
 DOCORDER
 DOCPICKINGLIST
 DOCRPTHEADER




PFKEYS: 1=HELP 6=PARMS 3=SAVE 12=CANCEL
```

Data Displayed

The Manage ObjectSet Permissions screen displays objects belonging to the object set for which access permissions are set. If accesses are not set for an object, it does not appear on the list. The objects are arranged by object type. You can vertically scroll individual lists by positioning the cursor beside the object type and pressing PF7 to scroll up or PF8 to scroll down.

Fetching Objects from the Object Set Definition

To retrieve any or all of the securable objects in the object set, enter the primary command **FETCH** in the primary command field of the Manage ObjectSet Permissions screen. You can specify an asterisk (*) to retrieve all objects or specify the object type (libraries, reports, screens, or tables) to retrieve objects of a specific type. For example:

```
Command ===> FETCH screens
```

The object fields in the screen for the object(s) you are retrieving must be empty before you can use this command.

Fetching Component Tables

Component screen tables and report tables are *not* automatically retrieved when you issue the **FETCH** command for screens or reports. All component tables that you want to secure must first be defined in the object set (they are not automatically included in the definition). To retrieve component tables, issue the **FETCH** command, specifying the table object type.

See Also *TIBCO Object Service Broker Application Administion* for details about the Object Set Definer.

Adding and Updating Permissions for Object Types

Adding, Changing, and Deleting Permissions

To add, change, or delete object permissions for each object type to be included, use the following PF keys to display a Specify Permissions screen:

PF4	Tables
PF5	Screens

PF6	Reports
PF9	Libraries

The Specify Permissions screen, illustrated below, displays existing object permissions. You also use it to specify:

- Other objects you want included in the permissions list
- Access modes that the object set requires for each object

Specify Table Permissions Screen

Use the Specify Table Permissions screen to specify permissions for a table object type. Each object type (library, report, screen, and table) has its own Specify Permissions screen. The layout of this screen is similar for each object type.

```
--------------------------------------------------------------------------------
        SPECIFY TABLE PERMISSIONS FOR OBJECTSET: DOCORDERENTRY
--------------------------------------------------------------------------------

       Table    Parms              Access Modes Required  TABLE
       Name           READ INS REPL DEL DEF_VIEW DEF_PRM MOD_DFN VIEW_DEFN CTRL
 --------------  -     -    -    -    -    -        -       -        -        -
 DOCBODY               Y    Y    Y    Y    Y        N       N        N        N
 DOCFOOTER             Y    Y    Y    Y    Y        N       N        N        N
 DOCHEADER             Y    Y    Y    Y    Y        N       N        N        N
 DOCORDERITEMS         Y    Y    Y    Y    Y        N       N        N        N
 DOCPICKINGLIST        Y    Y    Y    Y    Y        N       N        N        N
 DOCRPTHEADER          Y    Y    Y    Y    Y        N       N        N        N
 DOCUMENTMASTER        Y    Y    Y    Y    Y        N       N        N        N




 PFKEYS: 1=HELP 6=PARMS 3=SAVE 12=CANCEL
```

Specifying Objects and Accesses

To specify the objects to be included in your permissions list and the corresponding accesses, complete the following steps:

1. Type the object name(s).
2. Type the access permissions you want to assign.

For a complete description of the allowed access modes, refer to Task E: Set Accesses on page 72.

3. Specify access permissions for component screen tables and report tables.

   Component tables do *not* inherit the access permissions assigned to their parent.

4. Press PF3 to transfer the selected objects to the permissions list.

Specifying Parameter Values

If you include a parameterized data table, you can specify whether you are referring to either:

- Specific instances of the table
- The entire table

For example, you decide to allow wide access to the entire table and then restrict access to selected table instances (or vice versa). Your permissions list would then include permissions for the entire table as well as permissions for specific table instances.

If you specified the entire table, you can select any of the table access modes. If you specified a particular table instance, you can specify only modes that pertain to data access (that is, READ, INSERT, REPLACE, and DELETE).

Steps Required

For each parameterized table, complete the following steps:

1. Position your cursor beside the name of the parameterized table.
2. Press PF6 to invoke the Parameters screen.
3. Set ALL DATA to Y to refer to the entire table set (all instances).

   Or set **ALL DATA** to N and provide the parameters for the table instance you want to select.

4. Press Enter.

Saving Changes

To save this table specification, press PF3. The specification is not actually saved until you press PF3 from the Specify Permissions screen. Do *not* press PF12 to exit, unless you want to discard all the object permissions you just entered.

Specifying Control Permissions

If you specify control permissions in the object set permissions list, the following conditions apply:

- You can enable this object set only for user IDs; security groups cannot control objects.
- The object set can be enabled only by users who have ownership privilege on the objects for which control permission is given. Only the owner, the owner's security administrator, and a system administrator have ownership privilege on the object.

Adding and Updating User ID and Group Accesses

Who Can Specify User ID and Group Accesses?

Before you can enable an object set, you must specify which user IDs and groups should be permitted the accesses that you specified. You can update an object set membership list only if you have:

- Control permission on all the objects in the object set
- Ownership privileges to all the objects of the object set for which control permission is allowed

Invoking the Enable/Disable Object Set Screen

To specify user ID and group accesses to object sets, from the Security Management main menu position your cursor beside the **ObjectSet** field and do one of the following:

- Type in the name of an object set for which you have responsibility and press Enter.
- Press PF2 to display a list for selection. Type an S in the line command field next to the desired name and press PF3. When the selected name appears in the field, press Enter.

Using any of these methods causes the Enable/Disable Object Set screen, shown below, to appear.

Sample Enable/Disable Screen

From the Enable/Disable Screen you can create the list of user IDs or groups that you are allowing access to your object set. This screen also indicates if the object set is enabled or disabled. For more information about this state refer to Enabling and Disabling Object Sets on page 96.

```
--------------------------------------------------------------------------------
              Enable/Disable ObjectSet     DOCORDERENTRY
--------------------------------------------------------------------------------

NOTE: This objectset is currently DISABLED  ; will be  DISABLED  on SAVE

           _    Name                    User or Group
          (USERID | GROUP)               Description
          ----------------  ---------------------------------------
          STORES5G          ALL DEPARTMENT STORE LOCATIONS
          TELMKT12          TELEMARKETING - NORTHERN UNIT
          TELMKT18          TELEMARKETING - EASTERN









  PFKEYS: 1=HELP 6=USERIDS 9=GROUPS 21=VIEW 3=SAVE 5=ENABLE/DISABLE 12=CANCEL
```

Creating the Membership List

To create a membership list, do one of the following:

- Type the user ID and group names directly into the Name column.
- Press PF6 to select from a list of all available user IDs.
- Press PF9 to select a group name from a list of all security groups.

  You can position your cursor on the name of a group and press PF21 to view and select the users from a security group.

Deleting User ID and Group Access to Object Sets

To delete a user ID or group from the object set membership list, overtype the name with spaces or clear the field.

Saving Changes

Before you save the list, you must decide whether or not you want the object set enabled or disabled. For more information on enabling or disabling object sets, refer to Enabling an Object Set on page 96 and Disabling an Object Set on page 101.

To save changes, press PF3. This saves the list and exits the screen. To save any object set to be enabled, you must have at least one member listed.

Enabling and Disabling Object Sets

Enabling an Object Set

What is Enabling an Object Set?

After you specify access permissions to objects in an object set and you specify which user IDs and groups have access, the object set must be enabled to make those specifications effective. Enabling an object set is the process by which those pre-specified permissions are actually applied to the security access control list for objects.

An object set is enabled *directly* or *indirectly*, depending upon the existence of a parent-child relationship. If the object set being enabled has no children, it is enabled only for its own members and is said to be enabled directly.

To enable large object sets, consider increasing the value of the WORKINGSET Data Object Broker parameter.

Indirect Enabling of an Object Set

An object set can include another object set as its object. Indirect enabling occurs when a child object set is enabled through the enabling of its parent, *but only for the members of the parent*. This means that the same permissions specified for the members of the child object set are granted to members of the parent object set, when the parent is enabled.

Example

To illustrate this point, consider the following example:

Object Set	Table	Permissions	Members
(P) DOCORDERENTRY	DOCORDERITEMS	INSERT	STORES5G
(C) DOCORDERAUDIT	DOCAUDITITEMS	READ	PROJMGR8

In this example, P is the parent object set and C is the child. When the parent object set DOCORDERENTRY is enabled, the group STORES5G has insert access to DOCORDERITEMS and read access to DOCAUDITITEMS. However, PROJMGR8 does not have access to DOCAUDITITEMS since the child object set is not enabled for its own members.

Who Can Enable an Object Set?

You can enable an object set provided that *both* the following conditions are met:

- You have control access to all the objects in the object set. If a child object set is indirectly enabled, you must have security access to all its objects as well.
- You own all the objects for which control is given.

See Also *TIBCO Object Service Broker Parameters* for information about the WORKINGSET Data Object Broker parameter.

Methods Available to Enable an Object Set

Methods Available

There are two ways to enable an object set, interactively or in batch mode. Both methods are described in the following sections.

When an object set is enabled, whether directly or indirectly, lost permissions can result. For more information on lost permissions, refer to Effects of Enabling on Individual Object Permissions on page 99.

Enabling an Object Set Interactively

On all the platforms supported by TIBCO Object Service Broker, you can enable an object set interactively. To access the screen for enabling an object set, refer to Adding and Updating Permissions for Object Types on page 90.

On the Enable/Disable ObjectSet screen, the information line:

```
This objectset is currently DISABLED; will be DISABLED on SAVE
```

tells you the current status of the object set. The values displayed, ENABLED or DISABLED, indicate whether it is to be enabled or disabled when you use PF3 to exit from the screen.

Use PF5 to toggle between the enable and disable modes.

Enabling Object Sets Using Batch Processing

You can enable an object through the use of an interactive screen and a batch process.

Prerequisite

You must first have access to the @MAKEMEMBERS tool for this process. You can execute @MAKEMEMBERS without any additional setup if you use a level-7 user ID. If you are using a level-1 user ID, your user ID must first be added to the access list for the associated object set also named @MAKEMEMBERS and this object set must be enabled by a system administrator.

@MAKEMEMBERS takes the argument *object_set*. The value you provide for *object_set* is the name of the object set that is to be enabled. After executing @MAKEMEMBERS, a screen similar to the following appears:

```
--------------------------------------------------------------------------------
                Enable/Disable ObjectSet     DOCORDERENTRY
--------------------------------------------------------------------------------

NOTE: This objectset is currently DISABLED  ; will be  DISABLED  on SAVE

          _     Name                    User or Group
         (USERID | GROUP)                Description
         ----------------  ---------------------------------------
         STORES5G          ALL DEPARTMENT STORE LOCATIONS
         TELMKT12          TELEMARKETING - NORTHERN UNIT
         TELMKT18          TELEMARKETING - EASTERN









PFKEYS: 1=HELP 6=USERIDS 9=GROUPS 21=VIEW 3=SAVE 5=ENABLE/DISABLE 12=CANCEL
```

Steps to Enabling an Object Set

To enable the object set using this process, complete the following steps:

1. Execute the @MAKEMEMBERS tool to supply the names of the object sets and the names of the user IDs or groups who can use the object sets.
2. From the displayed screen, add users and groups in any of the following ways:

   You can add values by doing any of the following: typing in values, pressing PF6 to select from a list of all available user IDs, or pressing PF9 to select a group name from the list of all security groups.

   To save any object set to be enabled later via the BATCH_ENABLE tool, you must have at least one member listed.

3. Have your promotions administrator suspend users from the system.
4. Submit the BATCH_ENABLE tool for asynchronous processing, using the SCHEDULE statement from within a rule.

BATCH_ENABLE does the actual enabling of all object sets previously processed using @MAKEMEMBERS. In a z/OS environment, you can submit the BATCH_ENABLE tool to a queue using the BATCH or $BATCHOPT tools.

See Also

- *TIBCO Object Service Broker Managing Deployment* about suspending users.
- *TIBCO Object Service Broker Programming in Rules* about the use of the SCHEDULE statement.
- *TIBCO Object Service Broker Shareable Tools* about how to use the BATCH or $BATCHOPT tools to submit batch jobs.

Effects of Enabling on Individual Object Permissions

When an object set is enabled, you can no longer update the permissions to its component objects directly. This ensures that the integrity of the enabled object set is not compromised by changes in permissions to any of its individual objects.

Reporting Lost Permissions

When you enable an object set, you could receive a message indicating that permissions were lost when the object set was enabled. Use PF14 to display the report of lost permissions. Lost permissions are listed for those objects that are not included in other enabled object sets. They are permissions that were assigned directly to user IDs and groups and not through another enabled object set.

The information that you obtain by using PF14 is the only report you have on permissions lost in the enabling process. You can print it using PF13.

Re-establishing Permissions

Using the information on lost permissions obtained through PF14, you can re-establish lost permissions that are still required. To re-establish them, you can define other object sets and specify the permissions required.

Example

If application developer JONES lost his MOD_DFN rights to some payroll tables when the PAYROLL_READ object set was enabled, you could set up another object set called PAYROLL_MAINT. This second object set would provide JONES, and all other users requiring VIEW_DEFN and MOD_DFN access, with the ability to maintain the objects as required.

You cannot re-assign object permissions individually when they are part of an enabled object set. Refer to Updating Object Permissions When an Object Set is Enabled below for more information.

Saving Existing Permissions for Object Sets Enabled in Batch

When you enable an object set using BATCH_ENABLE you can retain the permissions for existing members of the object set as well as the members listed in the @MAKEMEMBERS table or you can delete the existing members and just enable it for the members listed in @MAKEMEMBERS.

Saving and Deleting When Using BATCH_ENABLE

BATCH_ENABLE uses the argument *wipe_existing*, which takes the values Y or N, to save or delete permissions for existing members:

- To save existing memberships, specify `BATCH_ENABLE('N')`.
- To delete existing memberships, specify `BATCH_ENABLE('Y')`.

Updating Object Permissions When an Object Set is Enabled

You can add to or delete user IDs or groups from the object set membership list, using the Enable/Disable screen:

- To add a user ID or group, press PF6 or PF9 and make the required selections.
- To delete a user ID or group, position your cursor on the user ID or group that you want to delete from the object set membership list and erase the name or overtype it with blanks.

After making the change, press PF3 to save.

To provide a user or group of users with a different set of permissions to objects used in the object set, you can create another object set that includes these objects and the required permissions. After creating this list, you can provide access by enabling the new object set.

Disabling an Object Set

What is Disabling an Object Set?

Disabling an object set is the process by which pre-specified permissions applied to the security access control list for objects through enabling are now deactivated.

To disable large object sets, consider increasing the WORKINGSET Data Object Broker parameter.

Who Can Disable an Object Set?

You can disable an object set provided that *both* the following conditions are met:

- You have control access to all the objects in the object set. If a child object set was indirectly enabled, you must have security access to all its objects as well.
- You own all the objects for which control is given.

Steps for Disabling an Object Set

To disable an object set for all users, complete the following steps:

1. Invoke the object set membership list as if you were enabling the object set.

   Refer to Invoking the Enable/Disable Object Set Screen on page 93 for further information.

2. Press PF5 to enable/disable the object set.

   The status line indicates that the object set is currently enabled and is disabled when you save.

3. Press PF3 to save the object set as disabled.

What Changes Can You Make?

After the object set is disabled, you can:

- Change individual object permissions for objects that are not part of any other enabled object set.
- Update the object set to include new objects, update the permissions, or delete an obsolete object.

After changes are made, you can enable the object set again. Refer to Enabling an Object Set on page 96.

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

Chapter 8 Auditing Accesses

This chapter describes how to audit the use of TIBCO Object Service Broker.

Topics

Auditing the Use of TIBCO Object Service Broker

As a system administrator, you could be required to audit the use of the TIBCO Object Service Broker system. Access to TIBCO Object Service Broker and its objects by a user is logged for auditing purposes in an audit log. Data in this audit log *cannot* be modified.

Using the Audit Log facility, you can query and report on your users' accesses to TIBCO Object Service Broker objects based on data stored in the audit log. Using purge tools, you can also archive the data stored in the log to files outside of TIBCO Object Service Broker.

What Determines the Level of Logging?

The SECAUDITLOG Execution Environment parameter determines the level of logging that is to take place. Security audit logging can be set to DISABLED, where nothing is recorded except for an initial entry indicating that SECAUDITLOG=DISABLED, or to NORMAL or STRICT, in which case more detailed security logging occurs. The default is NORMAL.

What is Logged in the Audit Log?

If the SECAUDITLOG parameter is set to NORMAL or STRICT, the audit log logs the following items:

- Security failures
- Security related events such as logins and logouts
- Changes to permissions or changing security groups
- All accesses to non-dictionary (MetaStor) tables when logging of user accesses is requested

If the SECAUDITLOG parameter is set to STRICT, all update accesses to non-dictionary (MetaStor) tables made by level-7 users are logged.

For more information about selectively logging user accesses, refer to Task D: Logging Accesses to a Table on page 70.

Who Can Access the Data in the Audit Log?

All level-7 users have read access to the audit log. All other levels of users must first be enabled to use the object set @AUDITLOG if they require access to the audit log data. Modifications to the audit log data are not allowed by any level of user. Refer to Chapter 7, Managing Object Set Security, on page 87 for information about enabling access to an object set.

Purging the Data Stored in the Audit Log

The data that is stored in the audit log should be purged on a regular basis. An archiving facility exists to assist you in this task, refer to Chapter 9, Archiving the Audit Log Data, on page 119 for information. The archiving mechanism and access to the external files that it uses are governed by external security.

Special Considerations for Strict Audit Logging

If the SECAUDITLOG Execution Environment parameter is set to NORMAL or STRICT, the following must be taken into consideration in your application development environment:

- The ACCESSLOG table used to store audit log data must be located on its own segment.
- Consider modifying the COMMIT points in applications. In a strict logging environment the number of COMMITS required for a level-7 user who updates a table is significantly higher than for a level-1 user.

See Also

TIBCO Object Service Broker Parameters for information about the SECAUDITLOG Execution Environment parameter.

TIBCO Object Service Broker for z/OS Utilities or *TIBCO Object Service Broker for Open Systems Utilities* about using the S6BBRIAL/hrnbrial (Move ACCESSLOG) utility to set up a segment for the audit log data.

TIBCO Object Service Broker Programming in Rules about committing changes.

Accessing the Audit Log Facility

Invoking the AUDITLOG Tool

To access the Audit Log facility, execute the AUDITLOG tool from the workbench:

```
EX execute rule ==> AUDITLOG<Enter>
```

You must have the appropriate security to use this facility. Refer to Who Can Access the Data in the Audit Log? on page 105 for a description of the security that you need.

Initial Screen of the Audit Log Facility

The first screen displayed, shown below, contains a list of filters available to you for querying the data. Reports can be created from these queries. If customized filters exist, they are also listed.

```
                    List of known filters for current user
 Command ==>                                                        Scroll P

          NAME          TYPE                 DESCRIPTION
    ---------------- -------- ---------------------------------------
  _ ACCESS_FAILED    DEFAULT  Access failed due to permission events
  _ ALL              DEFAULT  All events
  _ FAILURES         DEFAULT  Events where the access was not allowed
  _ LOGON            DEFAULT  Login and logout events
  _ LOGON_FAILED     DEFAULT  Login failure events
  _ OBJECT_ACCESS    DEFAULT  Accesses on all object types
  _ OBJECT_MAINT     DEFAULT  Object creation/deletion events
  _ PERMISSIONS      DEFAULT  Setting permission events
  _ SECURITY_LOG     DEFAULT  Archiving/Retrieving the Access log event
  _ TABLE            DEFAULT  All events on tables
  _ USER_GROUP       DEFAULT  All events on users and or groups




 D-Delete Filter  E-Edit Filter  Q-Query  R-Report
 PFKEYS: 12=EXIT 13=PRINT 3=END 5=FIND NEXT 9=RECALL
```

Available Options

From the initial Audit Log screen, you can:

- Query the audit log using a predefined filter
- Create a new filter
- Edit an existing filter that you defined
- Delete a filter that you defined
- Initiate a reporting session

These options are described in the following sections.

What is a Filter?

A filter is a predefined view of the audit data. For example, if you want to only view or report on data related to unsuccessful user accesses, you would use a filter that is specific to failures.

A number of filters are provided by default, as shown in the illustration above. The default filters cannot be edited or deleted but you can use a default filter as a template to create a new filter. If required, you can create additional filters from within this facility so that you can customize your auditing operations.

Querying the Data

A query of the audit log provides you with a tabular view of the data. This data is filtered through a selected filter so that only predefined occurrences appear.

To query the data, you can use a predefined filter or create a new filter that better suits your needs.

Using a Predefined Query Filter

To query a specific set of data, complete the following steps:

1. Type `Q` in the line command field of the filter that you want to use.
2. Press Enter.

   This displays a listing of accesses that meet the selection criteria specified in the filter, similar to the screen below.

Query Screen (first 80 columns shown)

```
COMMAND==>                                                      Scroll: P
--------------------------------------------------------------------------------
                        Audit Log - OBJECT_ACCESS
--------------------------------------------------------------------------------
                   Dates Ranging From: 20000310 To: 20000310


 Date    Time                          Message
------ -------- ------------------------------------------------------------
20000310 09:45:38 View definition access to TABLE IDM by USR51 denied
20000310 09:46:41 Create definition of TABLE IDM51 by USR51
20000310 09:48:08 View definition access to TABLE MSG_OLD_BKUP by USR52 denied
20000310 09:49:05 Create definition of TABLE BKUP_MESSAGES by USR50
20000310 09:53:47 Create definition of TABLE MSG_OLD_BKUP by USR50
20000310 09:54:06 Delete definition of TABLE IDM51 by USR51
20000310 09:55:52 Create definition of TABLE BKUP_MESSAGES by USR50
20000310 10:06:08 Replace access to TABLE FIELDS( @SLK_COLUMN ) by USR31
20000310 10:06:08 Replace access to TABLE FIELDS( @SLK_COLUMN ) by USR31
20000310 10:08:02 Replace access to TABLE @PROM_OBJ( LOC01 ) by USR01
20000310 10:08:02 Replace access to TABLE @PROM_INFO( LOC01 ) by USR01
20000310 10:08:03 Insert access to TABLE @PROM_AUDIT( LOC01 ) by USR01
20000310 10:08:03 Replace access to TABLE @PROM_OBJ( LOC01 ) by USR01
PFKEYS: 1=HELP 3=END 5=FIND NEXT 9=RECALL 12=END 13=PRINT 4=HIGHLIGHT FAILURES
```

Types of Data Displayed

Each filter is used to display a specific set of data. The initial display is sorted by date and time, with the current date displayed. You can browse the data, select, and re-order it to suit your requirements.

The screen displays the following types of information. Use PF11 to display the additional fields. This information is also used to produce the displayed messages.

Field	Information Supplied
Date	The date the access was logged.
Time	The time the access was logged.
Message	The message issued when the access took place. The message is in the following format: *Activity* of *Object_Type Object* by *User* [denied] [details]
Activity	The type of access the user was attempting, for example, a replace, logout, or view definition.
User	The user ID of the user performing the access.
User_Clearance	The security clearance of the user performing the access.
Object	The name of the object being accessed.
Object_Classfctn	The security classification of the object being accessed.
Object_Type	The type of object being accessed.
Access_Allowed	Specifies whether the user has permission to perform the access on the object.

Modifying the Display of Data

To modify the range of dates used for the data, enter new date ranges in the **Dates Ranging From** field, using the format *yyyymmdd*. You can also specify an asterisk (*) as a wildcard character to represent an entire date:

- An asterisk in the **Dates Ranging From** field means the earliest date in the access log.
- An asterisk in the **Dates Ranging To** field means yesterday's date.

A blank in either date field defaults to the current date.

To select and order this data, use the primary commands described when you press PF1. A description of the available PF keys is also provided when you press PF1.

When you are selecting on a specific field, ensure that you include the underscore character in a field with a compound name. For example, if you want to do a selection on Object Classification, specify **object_classfctn**.

Creating, Editing, and Deleting Filters

If you require a different set of criteria for a query or you want to create a new filter, initiate an editing session from the Audit Log screen. You can also delete a filter that you defined.

Creating a Filter

To create a new filter, complete the following steps:

1. Type **E** in the line command field of the filter that you want to use as a prototype.
2. Press Enter.

   This displays a screen similar to the one shown below, which you can edit.
3. Enter selection criteria directly into the Selection Criteria portion of the screen or select the criteria from the display in the bottom portion of the screen.

   When specifying selection criteria enclose character data in single quotes, for example: `OBJECT='DEPARTMENTS'`.
4. Enter a new description in the Description field.
5. Enter a new name in the Name field.
6. Press PF3.

   This saves the changes and returns you to the Audit Log screen.

Audit Log Filter Definition Screen

```
--------------------------------------------------------------------------------
                        Audit Log Filter Definition
--------------------------------------------------------------------------------
                     Name: OBJECT_ACCESS
               Description: Accesses on all object types
+------------------------------------------------------------------------------+
|       Selection Criteria                                                     |
|       ---------------------------------------------------------------        |
|       OBJECT_TYPE = 'LIBRARY' OR OBJECT_TYPE = 'REPORT' OR                   |
|       OBJECT_TYPE = 'SCREEN' OR  OBJECT_TYPE = 'TABLE'                       |
|                                                                              |
+------------------------------------------------------------------------------+
Fields of ACCESSLOG
_ TIME               _ USER              _ USER_CLEARANCE    _ OBJECT
_ OBJECT_CLASSFCTN   _ ACCESS_ALLOWED    _ MESSAGE_NO        _ PARM1
_ PARM2              _ ACTIVITY          _ OBJECT_TYPE
Relational Operator
_ =        _ <        _ >        _ <=       _ >=       _ ¬=       _ LIKE
Value / Expression

Logical Operator
_ AND     _ OR
PFKEYS: 1=HELP 3=SAVE 12=CANCEL 14=TBLS 15=RPTS 16=SCRS 17=LIBS 20=ACTIVITIES
```

Editing a User-Defined Filter

To edit an existing filter that you defined, use the same procedure that you used to create a filter, omitting step 5.

You cannot modify a default filter. To save editing changes that are made to a default filter, rename the filter before you save it.

Deleting a User-Defined Filter

You can delete filters that you defined, but not default filters. To delete one of your own filters, type **D** in the line command field beside the filter that you want to delete. After pressing Enter, you are prompted to confirm the deletion.

Samples of User Defined Selections

The following are examples of selection statements for user-defined filters:

- To track the activities of USR41, enter the following selection criteria:
  ```
  user='USR41'
  ```
- To track who has Insert/Delete/Replace access to the DEPARTMENTS table enter the following selection criteria:
  ```
  Activity='Insert' OR Activity='Delete' OR
  Activity='Replace')AND Object='DEPARTMENTS'
  ```
  For this example to work, accesses to the DEPARTMENTS table must be logged. For more information about logging accesses, refer to Task D: Logging Accesses to a Table on page 70.

Initiating a Reporting Session

If you want to run, edit, or delete reports, you can initiate a reporting session from the Audit Log screen.

Steps to Initiate a Reporting Session

To initiate a reporting session, complete the following steps:

1. From the List of known filters screen, type **R** in the line command field of the filter that you want to use.
2. Press Enter.

   This displays a screen similar to the one below.

List of Known Reports Screen

```
                    List of known reports for current user
Command ==>                                                        Scroll P

          NAME          TYPE                                          SUMMARY*
   ---------------- -------- ------------------------------------------------
 _ @ACCESS_BY_DATE  DEFAULT  Accesslog report broken down by date
 _ @ACCESS_BY_OBJCT DEFAULT  Accesslog report broken down by object name
 _ @ACCESS_BY_USER  DEFAULT  Accesslog report broken down by user
 _ @ACCESS_BY_USER2 DEFAULT  Accesslog report broken down by user
 _ @ACCESS_OBJ_DATE DEFAULT  Accesslog report broken down by object name and
 _ @ACCESS_OBJ_SUM  DEFAULT  Accesslog report broken down by object name with
 _ @ACCESS_USER_SUM DEFAULT  Accesslog report broken down by user with failur
 _ @ACCESS_USR_DATE DEFAULT  Accesslog report broken down by user and date




D-Delete  E-Edit  R-Run
PFKEYS: 12=EXIT 13=PRINT 3=END 5=FIND NEXT 9=RECALL
```

Running a Report

To run a report, complete the following steps:

1. From the List of known reports screen, type **R** in the line command field of the report that you want to use.
2. Press Enter.

   This displays a screen similar to the one below.

Audit Log Report Screen

```
--------------------------------------------------------------------------------
                              Audit Log Report

         Report: @ACCESS_BY_OBJCT - Accesslog report broken down by object nam
Report Criteria: OBJECT_ACCESS    - Accesses on all object types
--------------------------------------------------------------------------------


                    Report Dates from: 20000302 to: 20000318


          Report Destination:
            X Printer
              Message Log
              File:




PFKEYS: 1=HELP 3=END 2=LOG BROWSE 12=END ENTER=RUN REPORT
```

From this screen you can change the date range used for the report and send your report to a printer, the message log, or an external file. By default, the destination for the report is the printer defined for your session.

Modifying the Date Range

To modify the range of dates used for the data, enter new date ranges in the date range fields, using the format *yyyymmdd*. You can also specify an asterisk (*) as a wildcard character to represent an entire date:

- An asterisk in the **Report Dates from** field means the earliest date in the access log.
- An asterisk in the **Report Dates to** field means today's date.

Viewing the Report in the Message Log

If you want to view the report in the message log, complete the following steps:

1. Delete the **X** to the left of the Printer field.
2. Type **X** to the left of the Message Log field.
3. Press Enter.
4. Press PF2 after a message appears indicating that the report is ready.

If the report is too large to view in the Message Log, send the report to your printer or to an external file.

Sending the Report to an External File

To send your report to an external file, complete the following steps:

1. If you are operating in z/OS, pre-allocate a file.

   The file can be sequential or partitioned and must have Record Format FB, and LRECL of 120 or larger.

   This step is not required for the Open Systems platforms.
2. Delete the **X** to the left of the Printer field.
3. Type the name of the file to the right of the File field.

   If the file is a partitioned data set put the member name in brackets, for example, `FILE: SEC00.AUDIT.REPORT(PERMISSIONS)`

   If the file is a Windows or Solaris file, enter only the filename. The path is determined from your Open/Import/Export session option.
4. Press Enter.

Editing a User Defined Report Definition

If you require a different set of criteria for a report or you want to create a new report, initiate an editing session from the screen listing the reports. You *cannot* modify a default report. To save editing changes that are made to a default report, rename the report before you save it.

To edit a report definition, complete the following steps:

1. Type **E** in the line command field of the report that you want to use.
2. Press Enter.

This invokes the Report Definer with a definition of the selected report displayed, as shown below (first 80 columns shown).

Report Definer Screen

```
                        DEFINE REPORT: @ACCESS_BY_OBJCT       Unit:  NEWSEC
 Command ===>
 Page Length:  60   Page Width : 132  Fill Missing Data :


REPORTTABLES:       Origin  Max  Max   Title Blank  Title Heading Title Final
  Name              Row Col Occ  Acr   Only Overlap Rows  Rows    Cols  Cols
  ----------------  --- --- ---- ----   -      -     --    --      ---   ---
_ @ACCESS_BY_OBJCT    2   1    *    0   N            2     3        0     0
_ @ACCESSLOG_FILTR    1   1    1    0   Y            1     0        0     0
_ @ACCESSLOG_DATES   -2  50    1    0   Y            2     0        0     0
_
_
CONTROL BREAKS:     Max   Repeat   Title  Body              Control
  Name              Acr   Head     Rows   Reporttable       Break Field
  ----------------  ----  -        --     ----------------  ----------------
_                                  0     @ACCESS_BY_OBJCT  OBJECT
_                                  0     @ACCESS_BY_OBJCT
_                                  0

PFKEYS: 3=SAVE 12=CANCEL 2=DOC 22=DELETE 6=PAINT 13=PRT 21=DISPLAY 16=LITERAL
```

Deleting a User Defined Report

You can delete reports that you define but not default reports. To delete a report, complete the following steps:

1. Type **D** in the line command field of the report that you want to delete.
2. Press Enter.

You are prompted to confirm the deletion.

See Also *TIBCO Object Service Broker Defining Reports* about defining reports.

Chapter 9 Archiving the Audit Log Data

This chapter describes how to archive the audit log.

Topics

Archiving the Audit Log–z/OS

On an ongoing basis, the data in the audit log must be archived to an external file and deleted from the TIBCO Object Service Broker table. For security reasons, the only way to archive the data is to use either:

- PURGELOG_SCREEN, which is an interactive tool
- PURGELOG_BATCH, which is a batch tool

Conditions for Using the PURGELOG Tools

To run the PURGELOG tools, both the Data Object Broker and the Execution Environment must run on the same z/OS domain and be connected by z/OS Cross Memory Services (XMS). This ensures that the audit log data is protected by the resource-owning z/OS security definitions. To use XMS, both the Data Object Broker and the Execution Environment must run as Authorized Program Facility (APF) authorized programs.

In addition:

- The Execution Environment must be a single user environment.
- The object set @SEC_PURGELOG must be enabled for you.
- Adequate security must be defined externally for TIBCO Object Service Broker to enable you to run the PURGELOG routines and/or specify the external file that is used to store the archive data. Refer to Task A, Enter the name of the file where the audit data is to be stored, on page 126 for information about the purge file.

External Security Required to Archive Data

Through the z/OS security interface, System Authorization Facility (SAF), a RACROUTE call is made to a SAF-compliant security package such as CA-ACF2, RACF, or CA-Top Secret to verify access to the archive files.

To run the PURGELOG tools, two security definitions, *nodename*.SPECFILE and *nodename*.PURGELOG, must be defined to the external security system. The nodename is specified via the NODENAME Data Object Broker parameter.

These definitions are required so that a user(s) can:

- Specify the name of the archive file
- Run the archive process

Security Definition for Specifying an Archive File

nodename.SPECFILE must be specified as the name of the external security definition to which the user must be granted access before the user can specify the name of the archive file from within the PURGELOG tools. The definition must also identify the name of the TIBCO Object Service Broker node where the data is stored. It must be defined to the external security system in the form: `nodename.SPECFILE`.

Security Definition for Archiving the Data

nodename.PURGELOG must be specified as the name of the external security definition to which the user must be granted access before the user can archive the data from within the PURGELOG tools. The definition must also identify the name of the TIBCO Object Service Broker node where the data is stored. It must be defined to the external security system in the form: `nodename.PURGELOG`.

SAF Parameters Required

The following SAF parameters and values are defined for the RACROUTE macro definition. You must take these into consideration when defining the security definitions for your site:

TYPE=AUTH	The type of security request defined by TIBCO Object Service Broker
CLASS=S6BEE	The security class of the request defined by Execution Environment parameter SECCLASS.
USERID=*userid*	The TIBCO Object Service Broker user ID of the user initiating the archiving session
ENTITYX= *nodename.secdefn*	The name of the security definition that is to be verified. Depending on the action being preformed, the value is either *nodename*.PURGELOG or *nodename*.SPECFILE (where *nodename* is the name of the Data Object Broker that contains the data).
SUBSYS=S6B	The subsystem name as defined by the Execution Environment parameter SECSUBSYSTEM
REQUESTOR=S6BEEOP	The requestor name as defined by the Execution Environment parameter SECREQUESTOR

Samples Provided

Samples are provided with TIBCO Object Service Broker to assist you with preparing your security setup for your purge log functions. Your preparation depends on the external security package in use at your site. The following table lists the samples that are shipped with the CNTL data set distributed with TIBCO Object Service Broker:

Member Name	Description
SECEERAC	Sample for use with RACF.
SECEEACF	Sample for use with CA-ACF2.
SECEETSS	Sample for use with CA-Top Secret.

Archiving the Audit Log–Open Systems

On an ongoing basis, the data in the audit log must be archived to an external file and deleted from the TIBCO Object Service Broker table. For security reasons, the only way to archive the data is to use either:

- PURGELOG_SCREEN, which is an interactive tool
- PURGELOG_BATCH, which is a batch tool

Conditions for Using the PURGELOG Tools

To run the PURGELOG tools, adequate security must be defined for TIBCO Object Service Broker to allow you to run the PURGELOG tools and/or specify the external file that is used to store the archive data. Refer to Task A, Enter the name of the file where the audit data is to be stored, on page 126 for information about the purge file.

You must edit your secparm file, which is in the same directory as the crparm file, to specify the list of user IDs that are allowed to run these routines.

Required Entries for secparm

The secparm file contains three required entries:

userid	For users connecting to TIBCO Object Service Broker through an ostty session, specify the operating system user ID, not the TIBCO Object Service Broker user ID. It can be as long as 251 characters, but only the first 8 characters are used. For a user connecting through a TN3270 session, specify "3270". For a user connecting through theTIBCO Object Service Broker UI or the SDK (Java), specify "CLI".
specfile	A logical value (Y/y or N/n) indicating whether the *userid* is allowed to specify the archive file for the audit log
purgelog	A logical value (Y/y or N/n) indicating whether the *userid* is allowed to purge the audit log

If *userid* is “3270” or “CLI”, any user of the specified session type can archive the audit log.

Additional Content

Additional content must be as follows:

- The required entries can be separated by blanks or tabs.
- Lines beginning with the number symbol (#) are comment lines and are ignored.
- Blank lines are ignored.

Sample secparm File

```
# The following users are security administrators

# Format of the specifications are:
#
# <userid>    <specfile (Y/N)>    <purgelog (Y/N>
#
#USERA is allowed to specify the archive file but not allowed to purge the log
USERA Y N
#USERB is not allowed to specify the archive file but is allowed purge the log
USERB N Y
#USERXYZ is allowed to both specify the archive log and purge the log
USERXYZ Y Y
#The following specification is incomplete and is ignored
USERTEST Y
```

See Also *TIBCO Object Service Broker for z/OS Installing and Operating* or *TIBCO Object Service Broker for Open Systems Installing and Operating* for details about the crparm file.

Purging the Archive File Interactively

Invoking the PURGELOG_SCREEN Tool

To invoke PURGELOG_SCREEN from the workbench, execute the rule as follows:

```
EX execute rule ==> PURGELOG_SCREEN<Enter>
```

Prerequisite

You must have the appropriate security to use this facility. Refer to Archiving the Audit Log–z/OS on page 120 or to Archiving the Audit Log–Open Systems on page 123 for a description of the security that you require.

Purge Screen

```
--------------------------------------------------------------------------------
                        Archive and Purge Audit Log
--------------------------------------------------------------------------------

          Purge dates ranging from: 20070301 to: 20070310


     Archive File: S6B.AUDIT.LOG_____________________________________________




PFKEYS: 1=HELP 3=PURGE & EXIT 12=EXIT 4=VIEW ACCESSLOG
```

Use of the Purge Screen

Use this screen to complete these tasks, as described in the following sections:

1. Enter the name of the file where the audit data is to be stored, page 126

2. Supply the date range, page 127
3. Archive the data, page 127

Depending on your external security environment, one user could be required to enter the name of the archive file and another user could be required to archive the files.

Task A Enter the name of the file where the audit data is to be stored

If your user ID is defined to the SPECFILE security definition, enter the name of the file where the audit log data is to be stored. For example:

```
C:\S6B\Auditlog\Mar
```

or

```
S6B.AUDITLOG.MARCH
```

Press Enter to process the name of the file. If your user ID is defined to the PURGELOG security definition, you can then complete steps 2 and 3; otherwise, press PF12 to exit the screen.

Open Systems Requirements

If you are operating in a Windows or Solaris environment, enter the full path and filename. If the file does not already exist, it is allocated for you.

z/OS Requirements

In a z/OS environment, the following conditions apply for the file:

- Pre-allocate a file as follows: Record Format FB, LRECL 126 or larger. It can be a partitioned data set.
- If the file is partitioned, put the member name in brackets, for example:

  ```
  Archive File:SEC00.PURGE.LOG(JANUARY)
  ```

Task B Supply the date range

Specify the date range that is to be used. The dates must be supplied in the form *yyyymmdd*. You can also specify an asterisk (*) as a wildcard character to represent an entire date:

- An asterisk in the **Purge dates ranging from** field means the earliest date in the access log.
- An asterisk in the **Purge dates ranging to** field means yesterday's date.

The date range cannot include the current date; data for the current date cannot be purged.

Task C Archive the data

If your user ID is defined to the PURGELOG security definition, press PF3. This archives the data and returns you to the workbench.

Purging the Archive File in Batch

When the conditions described in Archiving the Audit Log–z/OS on page 120 or Archiving the Audit Log–Open Systems on page 123 are met, submit the PURGELOG_BATCH tool for asynchronous processing, using the SCHEDULE statement from within a rule. On z/OS, you can also submit PURGELOG_BATCH to a queue using the BATCH or $BATCHOPT tool.

Prerequisite

To use PURGELOG_BATCH, your user ID must be defined to *both* the SPECFILE and PURGELOG security definitions.

Usage of PURGELOG_BATCH

The arguments for PURGELOG_BATCH are:

fromdate	Supply the value using the same format described in Task B, Supply the date range, on page 127.
todate	Supply the value using the same format described in Task B, Supply the date range, on page 127.
file	Supply the value using the same format described inTask A, Enter the name of the file where the audit data is to be stored, on page 126.

See Also

TIBCO Object Service Broker Shareable Tools about how to use the BATCH or $BATCHOPT tools to submit batch jobs.

TIBCO Object Service Broker Programming in Rules about the use of the SCHEDULE statement and about running a rule in batch.

Analyzing Archived Data

The following technique may be used to analyze archived data on z/OS or Open Systems.

Two tables are provided:

1. @ARCH_ACCESSLOGI is an IMP table.

As supplied, it uses DD name ARCHIVE rather than a file name. If required, a level-7 user should be used to modify the definition of this table to use either a file name or DD name of your choice. The file or DD name should be specified according to instructions in "TIBCO Object Service Broker Managing External Data".

2. @ARCH_ACCESSLOG is a subview of the IMP table.

Browse this table to review the archived data.

For example, suppose data has been archived to data set 'S6B.ARCHIVE.DATA(D111007)':

- TSO ALLOC F(ARCHIVE) DA('S6B.ARCHIVE.DATA(D111007)') SHR REUSE

- Log on as a level-7 user

- BR @ARCH_ACCESSLOG

```
BROWSING TABLE   :  @ARCH_ACCESSLOG
COMMAND ==> SEL TIME >'06:08:39' & TIME < '06:08:42'

 IDKEY   DATE    TIME       MSG
_____________________________________________________________________________
 4       111007  06:08:40  Insert access to TABLE @OBJECTSETS( SYSADMIN ) by SYSADMIN
 5       111007  06:08:40  Insert access to TABLE @COMPONENTS( TESTCXEC ) by SYSADMIN
```

Chapter 10 Bound Security Access Data

This chapter describes how to bind the security access data and how to monitor the performance of the security.

Topics

Binding of Security Access Data

The Security Manager performs clearance checks on every access to every object. To perform clearance checks in an unobtrusive manner, the Security Manager copies object permissions into the Execution Environment and leaves them there for future reference.

This method of copying permissions, called binding, enhances performance. The Security Manager continues to refer to the memory-bound copy to validate accesses, until the Execution Environment is recycled or you refresh it with the latest information from the database.

Refreshing Bound Security Access Data

In most cases, security data can be refreshed without logging out of and in to TIBCO Object Service Broker by invoking a tool that rebinds the data.

If the object permissions are changed after they are bound, current users must refresh, or rebind, their security information by doing one of the following:

- Call the SEC_REBIND tool from a rule, as in the following example:

  ```
  CALL SEC_REBIND(NULL,NULL,NULL);
  ```

  The three arguments supplied with this tool, although not currently used, must be supplied. The example supplied three nulls. This enables a user to refresh the binding without logging out of the system.

If you access the Execution Environment in question through an OSB UI session, in order to pick up the rebound security data, be sure to also close and restart that OSB project after SEC_REBIND has been run.

- In a z/OS TSO environment, log out and log in again.
- In all other environments, shut down and restart the Execution Environment.

See Also *TIBCO Object Service Broker Application Administion* about binding.

TIBCO Object Service Broker Programming in Rules about rules.

TIBCO Object Service Broker for z/OS Installing and Operating or *TIBCO Object Service Broker for Open Systems Installing and Operating* about stopping and starting Execution Environments.

Monitoring Security Performance

Available Tool

Use the SECSTATS tool, available in the z/OS version of TIBCO Object Service Broker, to monitor security performance. With SECSTATS you can determine the optimal amount of memory to be used for bound security storage. The availability-to-demand ratios on various memory resident security control lists can be useful indicators of related storage usage performance.

See Also *TIBCO Object Service Broker for z/OS Monitoring Performance* about using SECSTATS.

Chapter 11 Password Encryption API

This chapter describes the password encryption facilities provided with TIBCO Object Service Broker.

Topics

Overview

Intended Audience

This chapter is meant for security administrators who are fully aware of the operational requirements for their security environment.

Facilities Available for Encryption

You can use the encryption application program interface (API) provided with TIBCO Object Service Broker to facilitate password encryption. This interface is implemented using an encryption loadable module.

Encryption Options Available

You can choose either of the following:

- No encryption, by using the default module provided with TIBCO Object Service Broker. Passwords are then left as clear text in the MetaStor.
- Encryption—either one-way or two-way—by replacing the default module with your own encryption loadable module. Sample code that demonstrates how to implement this option appears in install_path/sample/src/encrypt/sample.c for Open Systems, and, for z/OS, in the member HDRSCXIT in the ASM distribution data set supplied with TIBCO Object Service Broker. All components that participate in the storage or transmission of TIBCO Object Service Broker passwords use this interface.

Requirements

After you install your own custom encryption loadable module:

- The initial custom encryption loadable module must support the back-level encryption algorithm supplied with TIBCO Object Service Broker, which is clear text.
- You must customize all subsequent encryption loadable modules.

Limitations with One-way Encryption

If you choose to use your own encryption loadable module, the algorithm that you use for encryption can be one-way or two-way. If you choose to implement one-way encryption, the following TIBCO Object Service Broker facilities are unavailable to your users:

- @SCHEDULEMODEL table
- External security under IMS TM

These facilities require a clear text password and a one-way encryption prevents the decryption that is required to pass back the clear text password.

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

TIBCO Object Service Broker Application Administion about @SCHEDULEMODEL.

TIBCO Object Service Broker for z/OS External Environments about using IMS TM.

Encryption API for z/OS

Location of the API

The encryption API invokes the HDRSCXIT module. A combined default and sample version of this module is shipped with TIBCO Object Service Broker. The default version does not perform any encryption and the sample version performs a simple encryption algorithm.

As shipped in the ASM distribution data set, the module HDRSCXIT contains code, which could produce certain behavior, that is currently disabled. This code can be enabled by changing a constant in the module and re-assembling and re-linking it. To enable this behavior, refer to comments in the source code.

Supported Functions

The encryption API supports the following functions:

- Initialize
- Terminate
- Encrypt
- Decrypt
- VersionMismatch

For the syntax of each of these functions, refer to HDRSCXIT Module on page 139.

Initialize

Perform any necessary initialization when the Execution Environment is initialized. Return an address to a working storage memory block for use by subsequent encryption functions.

Terminate

Perform any cleanup necessary as a result of initialization when the Execution Environment shuts down.

Encrypt

Encrypt a clear text password that is supplied when a user logs in to the TIBCO Object Service Broker system. This encrypted password is compared to the value stored in the MetaStor. If the value is the same, the user can log in. If the value is different, the VersionMismatch function is invoked.

The encryption algorithm is at your discretionary control and can be one-way or two-way. The encrypted output must have a minimum length equal to the clear text input. The maximum encrypted length is specified by an input parameter.

Decrypt

Decrypt a given encrypted password. You must use this function for @SCHEDULEMODEL access and for external security under IMS TM.

If you choose to implement a one-way encryption algorithm, Decrypt always fails with a 0x04 return code.

VersionMismatch

Encrypt a clear text password using the previously supported algorithm. Compare this encrypted value to the previous value stored in the MetaStor for this user. If the two values are the same, the password in the MetaStor is updated with the value returned from the Encrypt function.

If the default encryption loadable module supplied with TIBCO Object Service Broker is replaced by a customer encryption loadable module, you must support, in the VersionMismatch function, clear text as the back-level encryption algorithm.

HDRSCXIT Module

Use the module HDRSCXIT, supplied as a member of the ASM distribution data set, as the sample for the customized API. After modifying the code, re-assemble HDRSCXIT and link it into your TIBCO Object Service Broker load library. It replaces the module shipped with TIBCO Object Service Broker.

Although it is at your discretion, we suggest that the functions ensure that the routine calling them is authorized to do so. To be authorized themselves, the functions must be re-entrant and linked as AMODE(31) RMODE(31).

Parameters

The parameter list for the API is a list of addresses pointed to by Register 1 (R1). All functions expect all parameters, even though some could be unused. The parameters are as follows:

Parameter	Length	Value
Function	Fullword	00 - Initialization call 04 - Encrypt call 08 - Decrypt call 12 - Password/Version mismatch 16 - Termination call
Site use	Fullword	A fullword to be used by the site.
Clear text password	Character, 8	The clear text password.
Length of clear text password	Fullword	The length of the clear text password.
Encrypted password	Character, 8	The encrypted password.
Length of encrypted password	Fullword	The length of the encrypted password.
Reserved	Fullword	Reserved for future use.

Special Considerations

Both the initialization call and the termination call occur only once per startup or shutdown of the Execution Environment. The other calls can occur many times in a multi-tasking environment such as CICS or a Native Execution Environment. Therefore, your exits must take the following points into consideration:

- Required resources, such as working storage, must be obtained and freed by each function as you determine it to be necessary.
- A resource obtained by the initialization call should be treated as READ only.

Encryption API for Open Systems

Location of the API

The external security API is defined in hrncrypt.h, located in the install_path/src/encrypt directory. Default and sample versions of the encryption loadable modules are shipped with TIBCO Object Service Broker. The default version performs no encryption and the sample version performs a simple encryption algorithm.

Location of the Loadable Modules

The following table lists the location required for the loadable modules after they are modified and re-compiled:

Platform	Loadable Module
Windows	install_path\bin\osencryp.dll
Solaris	install_path/sharedlib/liboscrypt.so

Required Compilers

The following table lists the compilers you must use to re-compile the hrncrypt module after you make changes to it:

Platform	C Compiler
Windows	Microsoft Visual C++ 2005.
Solaris	GCC 3.4.6 or later.

Sample

The install_path/src/encrypt directory contains a sample implementation of the encryption exit, in the file called sample.c.

Supported Functions

The encryption API supports the following functions:

- hrnEncryptInitialize
- hrnEncryptTerminate
- hrnEncrypt
- hrnDecrypt
- hrnVersionMismatch

For the syntax of these functions, see the sections below or refer to install_path/src/encrypt/hrncrypt.h file.

hrnEncryptInitialize

Performs any necessary initialization when the TIBCO Object Service Broker session is initialized. Returns an address to a working storage memory block for use by subsequent encryption functions.

```
int hrnEncryptInitialize(
      void** pStorage     /* OUT: Password encryption work area */
);
```

Parameters:

pStorage	Address of a pointer. This function uses this pointer to return the address of a work area to be passed in subsequent calls to hrnEncrypt, hrnDecrypt, hrnVersionMismatch, and hrnEncryptTerminate.

Return Values:

- HRN_ENCRYPTION_SUCCEED
- HRN_ENCRYPTION_FAIL

hrnEncryptTerminate

Cleans up, as necessary, any resources associated with the password encryption work area set up as a result of initialization. Also releases the storage for the area, if required, when the TIBCO Object Service Broker session terminates.

```
void hrnEncryptTerminate(
         void* pStorage          /* IN: Password encryption work area */
);
```

Parameters:

pStorage	Address of a pointer to the work area created by the first hrnEncryptInitialize call.

hrnEncrypt

Encrypts a clear text password that is supplied when a user logs in to the system. This encrypted password is compared to the value stored in the MetaStor. If the value is the same, the user can log in. If the value is different, the hrnVersionMismatch function is invoked.

The encryption algorithm is at your discretionary control and can be one-way or two-way. The encrypted output must have a minimum length equal to the clear text input. The maximum encrypted length is specified by an input parameter.

```
int hrnEncrypt(
      void*    pStorage                 /* IN: Password encryption work area */
      encrypt_byte_t* pClearText,       /* IN: Clear text password */
      encrypt_size_t wClearTextLen,     /* IN: Actual clear text password length */
      encrypt_byte_t* pEncrypted,       /* OUT: Encrypted text */
      encrypt_size_t* pwEncryptedLen,   /* IN/OUT: Length of encrypted text*/
      encrypt_size_t* pwDummy            /* N/A */
);
```

Parameters:

pStorage	Address of a pointer to the work area.
pClearText	Address of the clear text password entered by the user logging in.

wClearTextLen	Length of the clear text password string.
pEncrypted	Address of area to return the encrypted password.
pwEncryptedLen	In: The length of the area for the encrypted password. Out: The length of the resulting encrypted password.
pwDummy	Reserved for future use.

Return Values:

- HRN_ENCRYPTION_SUCCEED
- HRN_ENCRYPTION_FAIL

hrnDecrypt

Decrypts a given encrypted password. You must use this function for @SCHEDULEMODEL access.

If you chose to implement a one-way encryption algorithm, hrnDecrypt always fails with a 0x04 return code.

```
int hrnDecrypt(
      void*    pStorage                  /* IN: Password encryption work area */
      encrypt_byte_t* pEncrypted,        /* IN: Encrypted password text */
      encrypt_size_t wEncryptedLen,      /* IN: Esncrypted password text length */
      encrypt_byte_t* pClearText,        /* OUT: Clear text password */
      encrypt_size_t* pwClearTextLen     /* IN/OUT: Length of clear text password*/
);
```

Parameters:

pStorage	Address of a pointer to the work area.
pEncrypted	Address of the encrypted password to be converted to clear text.
wEncryptedLen	Length of the encrypted password string.
pClearText	Address of the area to return the clear text password.

pwClearTextLen	In: The length of the area for the clear text password. Out: The length of the resulting clear text password.

Return Values:

- HRN_ENCRYPTION_SUCCEED
- HRN_ENCRYPTION_FAIL

hrnVersionMismatch

Encrypts a clear text password using the previously supported algorithm. Compares this encrypted value to the previous value stored in the MetaStor for this user. If the two values are the same, the password in the MetaStor is updated with the value returned from the hrnEncrypt function.

If the default encryption loadable module supplied with TIBCO Object Service Broker is replaced by a customer encryption loadable module, in the hrnVersionMismatch function you must support clear text as the back-level encryption algorithm.

```
int hrnVersionMismatch(
      void*    pStorage                 /* IN: Password encryption work area */
      encrypt_byte_t* pClearText,       /* IN: Clear text password */
      encrypt_size_t wClearTextLen,     /* IN: Actual clear text password length */
      encrypt_byte_t* pEncrypted,       /* OUT: Encrypted text */
      encrypt_size_t* pwEncryptedLen,   /* IN/OUT: Length of encrypted text*/
      encrypt_size_t* pwDummy           /* N/A */
);
```

Parameters:

pStorage	Address of a pointer to the work area.
pClearText	Address of the clear text password entered by the user logging in.
wClearTextLen	Length of the clear text password string.
pEncrypted	Address of the area to return the encrypted password.
pwEncryptedLen	In: The length of the area for the encrypted password. Out: The length of the resulting encrypted password.

pwDummy	Reserved for future use.

Return Values:

- HRN_ENCRYPTION_SUCCEED
- HRN_ENCRYPTION_FAIL

Changing the Encryption Algorithm

Steps Required for z/OS

When you make changes to the encryption algorithm, you must re-assemble the encryption loadable module and re-link into the load library. All TIBCO Object Service Broker components, except the Data Object Broker, must be stopped and restarted before the new algorithm takes effect.

Steps Required for Open Systems

When changes are made to the encryption algorithm, you must re-compile the encryption loadable module and replace it in the bin directory for Windows and the sharedlib directory for Solaris. All TIBCO Object Service Broker components, except the Data Object Broker, must be stopped and restarted before the new algorithm takes effect.

How Do Multiple Changes of an Algorithm Affect a User?

If the encryption algorithm changes twice between a user logging in (usually because of an extended absence), the user is locked out of TIBCO Object Service Broker. A system administrator or the user's security administrator must then assign a new password to the user ID. Refer to Chapter 3, Managing User Profiles, on page 25 for information on setting passwords in a user profile.

Automatically Upgrading Passwords

When the encryption algorithm changes, the VersionMismatch or hrnVersionMismatch function automatically upgrades passwords. Refer to Encryption API for z/OS on page 138 for information about the VersionMismatch function and Encryption API for Open Systems on page 141 for information about the hrnVersionMismatch function.

If a user is successful at doing a back-level encryption from the S6BBRULH/hrnbrulh (Batch Unload (Online)) utility, the login is successful and an automatic upgrade is not performed. A warning message appears, to indicate that an upgrade is pending. To cause the upgrade to take effect, the user must log in interactively with the appropriate user ID.

See Also *TIBCO Object Service Broker for z/OS Installing and Operating* or *TIBCO Object Service Broker for z/OS Installing and Operating* for information about stopping and starting TIBCO Object Service Broker components.

TIBCO Object Service Broker for z/OS Utilities for your operating environment about the S6BBRULH/hrnbrulh utilities.

Chapter 12 Implementing External Security

This chapter describes how to implement external security with TIBCO Object Service Broker.

Topics

Overview

Intended Audience

This chapter of the manual is meant for security administrators who are fully aware of the operational requirements for their security environment, including the interfaces to the security packages in use at their sites.

When Does TIBCO Object Service Broker Use External Security?

TIBCO Object Service Broker uses external security to control access to the Data Object Broker. This includes the use of the S6BTLADM/hrntladm (Administration Menu) utility, some operator commands, and some utilities. Refer to, Defining Data Object Broker Access, on page 153 for more information.

TIBCO Object Service Broker always uses external security when the audit log generated by TIBCO Object Service Broker security is being archived. Refer to Chapter 9, Archiving the Audit Log Data, on page 119 for more information.

Your TIBCO Object Service Broker system can also be set up to use external security to verify user ID login. This security is in effect at your Execution Environment and session level. Refer to Defining User Validation on page 158 for more information.

How Does TIBCO Object Service Broker Invoke External Security for Logins?

In TIBCO Object Service Broker, you can set up your processing environment to pass control from TIBCO Object Service Broker security to the native security for your operating environment or an external security provider of your choice. As described in Chapter 2, Security Clearances, on page 11, setting the SECURITY Execution Environment parameter to EXTERNAL initiates external security processing. If the parameter is set to MIXED, first TIBCO Object Service Broker security is checked and then, if the access does not pass validation, external security processing is initiated.

What is the Default Implementation?

The default implementation supplied with TIBCO Object Service Broker is SECURITY=INTERNAL. This means that only TIBCO Object Service Broker security is used for user ID login verification.

Effects on Data Accesses to Peer TIBCO Object Service Broker Nodes

If SECURITY=EXTERNAL is specified, no additional security is required when a user accesses peer TIBCO Object Service Broker nodes. Both sites do not have to be using the same external security package. If a login is successful at a local node with external security enabled, it is successful at a remote node that has external security enabled.

What External Security Interfaces are Supported?

The following table lists the supported external security interfaces:

Platform	Description
z/OS	System Authorization Facility (SAF) interface: this interface is used by security packages such as RACF, CA-ACF2, and CA-Top Secret.
Windows	Windows native security • Generic Security Service (GSS) API: this interface is used by security packages such as Kerberos or Sesame, and by custom security exits. • Lightweight Directory Access Protocol (LDAP) API: this interface can be used to query centralized directory servers.
Solaris	UNIX native security • GSS API: this interface is used by security packages such as Kerberos or Sesame, and by custom security exits. • LDAP API: this interface can be used to query centralized directory servers.

Reference Information

GSS API

X/Open Preliminary Specification: Generic Security Service API (GSS API)

RFC 1508: Generic Security Service Application Programming Interface

RFC 1509: Generic Security Service API: C-bindings

z/OS SAF interface

IBM RACF documentation or documentation for the external security package your site is using.

See Also *TIBCO Object Service Broker Application Administion* about the security requirements for data accesses to peer nodes if TIBCO Object Service Broker security is in use.

TIBCO Object Service Broker for z/OS Installing and Operating or *TIBCO Object Service Broker for Open Systems Installing and Operating* about the Administration menu and operator commands.

TIBCO Object Service Broker Parameters about Execution Environment parameters.

Defining Data Object Broker Access

What Accesses are Affected?

The following functions and utilities are affected by the external security setup for your Data Object Broker:

- The S6BTLADM/hrntladm (Administration Menu) utility
- The S6BTLCMD (Submit Operator Commands in Batch) utility used to issue operator commands in batch on z/OS
- The hrnspset (Reset Journal) utility used to reset a full journal so that it can be reused by TIBCO Object Service Broker on Open Systems
- The S6BSPJEX (Journal Data Extraction) utility used to extract records collected in journal data sets and copy them to a data set on z/OS
- The S6BTLBRM (Resource Management Online Backup) utility used to create a flat file backup of the resource data stored in the resource repository on z/OS
- The hrncr (Data Object Broker) utility used to start, maintain, and stop the Data Object Broker on Open Systems

When using the Administration menu or issuing operator commands, a user's user ID is validated against its classification level as defined for external security.

How are the Accesses Specified for z/OS?

On z/OS, you use the System Authorization Facility (SAF) and a SAF-compliant package such as CA-ACF2, RACF, or CA-Top Secret to verify accesses to the Data Object Broker. Refer to External Security Interface–z/OS on page 155 for information on how to define a user ID with a specific classification level.

Classifications Levels–z/OS

The following table describe, in increasing authority, the user classification levels available for z/OS. Press PF11 in the Administration menu for a full listing of the accesses available based on user classification.

Classification	Description
General	Can view statistical information only. There are no update privileges.
Privileged	Can view a more complete set of statistical information than a general user. There are no update privileges.
Administrator	In addition to the accesses that a Privileged user has, an Administrator user can also define and manage resources and issue update operations commands against the Data Object Broker.
Operator	In addition to the accesses that an Administrator user has, an Operator user can also access all the displays and functions of the Administrator, including the high impact operator commands.

How are the Accesses Specified for Open Systems

On Open Systems, you define user IDs to a specific classification via the PRIVILEGED, OPERATOR, and SYSADMIN Data Object Broker parameters. Refer to *TIBCO Object Service Broker Parameters* for details on how to define these parameters.

User Classifications–Open Systems

This table describes the user classifications available for Open Systems.

Classification	Description
General	Can view statistical information only from the Administration menu. There are no update privileges. All users have General access.
Privileged	Can view a more complete set of statistical information than a general user. There are no update privileges. A maximum of 15 users can be defined as Privileged.

Classification	Description
Operator	In addition to the accesses that a Privileged user has, an Operator user can also issue operations commands against the Data Object Broker. A maximum of 15 users can be defined as an Operator.
Sysadmin	Access to all the displays and functions of the Administration menu, including the operator commands. Only *one* user can be defined as Sysadmin.

External Database Servers

For implementation details about the external database servers, refer to the appropriate Service Gateway manual in the TIBCO Object Service Broker documentation.

See Also

TIBCO Object Service Broker for z/OS Installing and Operating or *TIBCO Object Service Broker for Open Systems Installing and Operating* for details on the Administration menu and operator commands.

TIBCO Object Service Broker for z/OS Utilities or *TIBCO Object Service Broker for Open Systems Utilities* for details about the utilities.

TIBCO Object Service Broker Parameters for details on how to define the SECUREADMIN parameter.

External Security Interface–z/OS

What is the Interface?

The z/OS security interface, System Authorization Facility (SAF), is used to verify accesses to the Data Object Broker. SAF is enabled by setting the SECURADMIN Data Object Broker parameter to Y. By default this parameter is set to N (disabled). If security is enabled, a RACROUTE call is made to a SAF-compliant security package such as RACF, CA-ACF2, or CA-Top Secret.

External Security Requirements

You must define the following three security definitions:

- *nodename*.PUSER

- *nodename*.ADMIN
- *nodename*.OPER

where *nodename* refers to the Data Object Broker that contains the data to be accessed. Nodename is specified via the NODENAME Data Object Broker parameter.

You define these definitions for each Data Object Broker at your site that has the SECUREADMIN Data Object Broker parameter set to Y. The default access should be NONE; READ access should be given to the user ID being granted access at the specified level.

SAF Parameters Required

The following SAF parameters and values are defined for the RACROUTE macro. You must take these into consideration when defining the security definitions for your site:

TYPE=AUTH	The type of security request defined by TIBCO Object Service Broker
CLASS=S6BDOB	The security class of the request defined by the Data Object Broker parameter SECCLASS
USERID=*userid*	The user ID of the requestor
ENTITY=*nodename.secdef*	The name of the security definition that is to be verified. The value is one of: *nodename*.PUSER, *nodename*.ADMIN, or *nodename*.OPER
SUBSYS=S6B	The subsystem as defined by TIBCO Object Service Broker
REQUESTOR= S6BDOBOP	The internal requestor name as defined by the Data Object Broker parameter SECREQUESTOR

Samples Provided

Samples are provided with TIBCO Object Service Broker to assist you with preparing your security setup for your administration functions. Your preparation depends on the external security package in use at your site. The following table lists the samples that are shipped with the CNTL data set:

Member Name	Description
SECCORAC	Sample for use with RACF.
SECCOACF	Sample for use with CA-ACF2.
SECCOTSS	Sample for use with CA-Top Secret.

See Also *TIBCO Object Service Broker for z/OS Installing and Operating* or *TIBCO Object Service Broker for Open Systems Installing and Operating* for details on the Administration menu and the operator commands.

Defining User Validation

Implementation Requirements for User Validation

The operational and security interface environment where you are working determines your implementation requirements in terms of user ID and password validation:

- z/OS makes use of a SAF interface, which in turn is used by other security packages.
- Open Systems provide their own default security, which can be replaced by other security packages.
- GSS and LDAP provide generic interfaces that can be used by Open Systems.
- Each external database server has its own unique requirements.

z/OS

When external security is enabled, user validation is automatically provided through your external security package. Other than specifying SECURITY=EXTERNAL or MIXED as an Execution Environment parameter, no additional steps are required.

The use of External Security for User Validation in a Native Execution Environment requires that the S6BDR000 program be defined to the External Security package (e.g., CA-Top Secret or CA-ACF2) with a multi-user address space attribute.

In CA-Top Secret this is accomplished via the Multi User Single Address Space Subsystem (MUSASS) option and in CA-ACF2 via the Master Facility option.

Consult the appropriate documentation if you are using some other External Security package.

Windows

Windows systems can validate user IDs locally or through a domain. If your environment is using domain security, TIBCO Object Service Broker uses the osee.exe process to determine the name of the domain to be queried.

Providing Adequate Security Privileges

To give a user ID adequate privileges to use osee.exe, complete the following steps:

1. Select Start > Programs > Administrative Tools from the Task bar.
2. Select User Manager > Policies > User Rights…
3. Click the check box for “Show Advanced User Rights”.
4. From the Right list box select “Act as part of the operating system”.
5. Click on Add…
6. From the Add Users and Groups window, select the required user ID.

   These are the user IDs under which osee.exe is to execute.
7. Click OK to exit the Add Users and Group window and User Rights Policy window.
8. Exit the User Manager window.
9. Reboot the machine to enable the changes.

Additional Requirements

The following are also required:

- If using domain security, the domain controller (or its backup) must be operational.
- A user’s Windows password must be limited to eight characters.

Solaris

Solaris uses the getspnam() function to return a user ID and password for comparison with values supplied by a user. The encrypted passwords for users are stored in a shadow file (/etc/shadow). (Both the getspnam() function and the shadow file must be accessible to a super-user.)

The calling program, hrnsecur, must have an effective user ID of the super-user to obtain a secured encrypted password. To set this, complete the following steps:

1. Use the chown command to set the owner of hrnsecur to root.
2. Use the chmod u+s command to set the set-user-id bit.

External Database Servers

For implementation details about the external database servers, refer to the appropriate Service Gateway manual in the TIBCO Object Service Broker documentation.

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

External Security API–Open Systems

Location of the API

The external security API is declared in hrnsecur.h, in the install_path/src/security directory.

Required Compilers

Platform	C Compiler
Windows	Microsoft Visual C++ 2010.
Solaris	GCC 3.4.6 or later.

Supported Functions

The external security API supports the following functions:

- hrnExtSecClientInit
- hrnExtSecValidateUser

For the syntax of these functions, refer to the sections below or to install_path/src/security/hrnsecur.h file.

hrnExtSecClientInit

Some security providers require the use of an opaque token to authenticate a client to a server. Using this function, the external security provider can create this token. The client caller provides an Execution Environment name, a user ID, and a password. The exit returns a pointer to a token and a length. It is not necessary for the external security provider to use any of the returned information if a token does not have to be built. The created token is specific to the security provider and is not interpreted or used by TIBCO Object Service Broker.

```
int hrnExtSecClientInit(
      char*    pszEEName,      /* IN: ASCII EE Name */
      char*    pszUserid,      /* IN: ASCII Userid */
      char*    pszPassword,    /* IN: ASCII Password */
      int*     pnTokenLen,     /* OUT: Token length */
      void**   pToken          /* OUT: Security token */
);
```

Parameters:

pszEEName	Address of a null-terminated ASCII string containing the Execution Environment name.
pszUserid	Address of a null-terminated ASCII string containing the TIBCO Object Service Broker user ID entered by the client.
pszPassword	Address of a null-terminated ASCII string containing the TIBCO Object Service Broker password entered by the client.
pnTokenLen	Address of an integer containing the length of the security token to be sent to the Execution Environment.
pToken	Address of the security token to be sent to the Execution Environment. The hrnExtSecValidateUser function uses this token to authenticate session login.

When using the SDK (C/C++), you must always supply a password among the STARTSS session parameters for a STARTSS cliProc request. If you use external security to avoid supplying a password, use a dummy value for the PASSWORD parameter.

Return Values:

- HRN_EXTSEC_SUCCEED
- HRN_EXTSEC_FAIL – the security provider detects an error; the session terminates with a general login failure
- HRN_EXTSEC_NOTIMPLEMENTED – the exit is not implemented

hrnExtSecValidateUser

This function, given a user ID, password, and token, determines if the user is allowed to log in to the system. The hrnExtSecValidateUser exit is called when the SECURITY Execution Environment parameter is set to EXTERNAL or MIXED.

```
int hrnExtSecValidateUser(
      char*    pszUserid,          /* IN: ASCII Userid */
      char*    pszPassword,      /* IN: ASCII Password */
      char*    pszEEName,        /* IN: ASCII EE Name */
      void*    pToken,             /* IN: Token */
      void**   pSecurityEnv     /* RESERVED */
);
```

Parameters:

pszUserid	Address of null-terminated ASCII string containing the TIBCO Object Service Broker user ID entered by the client.
pszPassword	Address of null-terminated ASCII string containing the TIBCO Object Service Broker password entered by the client. This can be an empty string if authentication information is being passed via a security token.
pszEEName	Address of a null-terminated ASCII string containing the osMon name.
pToken	Address of the security token created by the hrnExtSecClientInit exit invoked for the current client and passed on to the Execution Environment during session login processing.
pSecurityEnv	Reserved for internal use.

The length of *pToken* is not passed as an argument. You should either pass tokens of a predetermined size or embed the length within the token. If you choose to embed the length within the token, you must manage any byte-ordering issues, such as big vs. little endian type.

Return Values:

- HRN_EXTSEC_SUCCEED

- HRN_EXTSEC_FAIL – the security provider detects an error; the session terminates with a general login failure
- HRN_EXTSEC_BADUSERID – the user ID is invalid; this error is passed on to the client
- HRN_EXTSEC_BADPASSWORD – the password is invalid; this error is passed on to the client
- HRN_EXTSEC_NOTIMPLEMENTED – the exit is not implemented

Source of the External Security Loadable Modules

Location of the Source Files

You can find the source files used to implement the default Windows, Solaris, LDAP, and GSS modules in the install_path/src/security folder, as shown here:

Platform	Source File	Implementation
Open Systems	default.c	Default.
Open Systems	gss.c	GSS.
Open Systems	ldap.c	LDAP.
Windows	nt.c	Native security.
Solaris	unix.c	Native security.
Solaris	agent.c	UNIX agent program.

In the install_path/src/security folder, the hrnsecur.h file contains the declarations of the API contained in the source files in that folder.

Location of the Loadable Modules

The following table lists the location for the modules required for a default security implementation and the sample modules required to create a customized loadable module for your site:

Platform	Loadable Module	Implementation
Windows	install_path\bin\ossecur.dll	Default.
Solaris	install_path/sharedlib/libossecur.so	Default.

Index

Symbols

A

B

C

D

E

F

G

H

I

L

M

N

O

P

R

S

T

U

V

W