

# **TIBCO® Object Service Broker**

## **Defining Screens and Menus**

*Software Release 6.0*

*July 2012*

## Important Information

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

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

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

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

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

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

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

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

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

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

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

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

TIBCO Software Inc. Confidential Information

# Contents

<b>Preface</b> .....	<b>ix</b>
Related Documentation .....	x
TIBCO Object Service Broker Documentation .....	x
Typographical Conventions .....	xv
Connecting with TIBCO Resources .....	xviii
How to Join TIBCOCommunity .....	xviii
How to Access All TIBCO Documentation .....	xviii
How to Contact TIBCO Support .....	xviii
<b>Chapter 1 Introduction to TIBCO Object Service Broker Screens</b> .....	<b>1</b>
TIBCO Object Service Broker Screens .....	2
What is a Screen? .....	2
What is a Screen Table? .....	2
Advantages of Using Screen Tables .....	2
TIBCO Object Service Broker Menus .....	3
What is a Menu? .....	3
Menu Styles .....	3
Validation .....	4
Phase One Validation .....	4
Phase Two Validation .....	4
Validation Failure .....	4
<b>Chapter 2 Building a Screen</b> .....	<b>7</b>
The Screen Definer .....	8
Accessing the Screen Definer .....	8
Layout .....	8
Example .....	8
Tasks for Defining a Screen .....	9
Task A: Verify Screen Identification .....	10
Purpose of this Task .....	10
The Define Screen and Unit Fields .....	10
Task B: Specify Screen Tables .....	11
Purpose of this Task .....	11
Defining Required Screen Table Attributes .....	11
Defining Optional Screen Table Attributes .....	13
Excluding Screen Tables .....	14

Task C: Enter PF Key, Scrolling, and Cursor Information . . . . .	15
Purpose of this Task . . . . .	15
Assign PF Key Values . . . . .	15
Define the Scroll Amount Entry . . . . .	16
Define the Default Cursor Position. . . . .	17
Task D: Specify PF Keys. . . . .	18
Assigning PF Keys. . . . .	18
Specifying Function Key Attributes . . . . .	18
@TMP_FCNKEYS Table Example . . . . .	19
Deleting a Screen Definition . . . . .	20
Considerations when Deleting a Screen . . . . .	20
Deleting a Definition Using the Screen Definer . . . . .	20
Deleting a Definition Using a Shareable Tool. . . . .	21
Maximizing Screen Layout . . . . .	22
Screen Table Definitions . . . . .	22
Last Row Attribute Set to 22 . . . . .	23
Last Row Attribute Set to -2. . . . .	24
<b>Chapter 3 Defining Screen Tables . . . . .</b>	<b>25</b>
Screen Tables and Screen Fields . . . . .	26
Screen Tables . . . . .	26
Screen Fields. . . . .	26
The Screen Table Painter . . . . .	27
Invoking the Screen Table Painter . . . . .	27
Using the Screen Table Painter. . . . .	28
Layout . . . . .	28
Screen Table Painter Screen . . . . .	28
Image Area . . . . .	29
Field Definition Area. . . . .	29
Available PF Keys for the Screen Table Painter. . . . .	30
Defining Screen Tables . . . . .	30
Adding Literal Text. . . . .	31
Definition . . . . .	31
Creating Literal Text. . . . .	31
Setting Display Attributes. . . . .	32
Display Attributes for Literals . . . . .	32
Literal Attributes Screen. . . . .	33
Defining Data Fields . . . . .	34
Methods for Defining Data Fields. . . . .	34
Benefits of Using Predefined Fields. . . . .	34
Benefits of Adding Fields with Default Attributes . . . . .	35
Benefits of Creating New Fields. . . . .	35

Using Predefined Fields. . . . .	36
Using Table Field Definitions . . . . .	36
Using Global Fields . . . . .	38
Selecting a Global Field . . . . .	38
Using Fields with Default Attributes . . . . .	40
Creating Screen Fields with Predefined Attributes . . . . .	40
Creating New Fields. . . . .	41
Data Fields . . . . .	41
Determining How the Data is Presented . . . . .	42
Determining How a Field Operates. . . . .	43
Controlling Presentation. . . . .	44
Determining Uses for a Field . . . . .	44
Determining Appearance . . . . .	45
Expanding Definitions . . . . .	47
Literal Text . . . . .	47
Field Definitions . . . . .	47
Expanding a Field. . . . .	48
Editing Screen Tables and Screen Fields . . . . .	49
Adding a Line or Field. . . . .	49
Copying a Line or Literal. . . . .	49
Cutting a Field or Literal . . . . .	49
Pasting a Field, Line, or Literal . . . . .	49
Deleting a Line or Field. . . . .	50
Deleting a Screen Table . . . . .	50
<b>Chapter 4 Adding Additional Features to a Screen . . . . .</b>	<b>53</b>
Adding Help Information . . . . .	54
Overview of Help . . . . .	54
Adding Screen Help . . . . .	54
Examples of Screen Help. . . . .	54
Adding Field Help . . . . .	56
Examples of Field Help. . . . .	57
Color Implementation. . . . .	60
Adding Color. . . . .	60
Purpose of the @COLOURS Table . . . . .	60
Fields in the @COLOURS Table . . . . .	61
Purpose of the @SCREENCOLOURS Table . . . . .	61
Fields in the @SCREENCOLOURS Table . . . . .	62
Adding Virtual Fields to a Screen Table . . . . .	63
Using Virtual Fields. . . . .	63
\$MODIFIED Field . . . . .	63
\$OCCNUM Field. . . . .	64

<b>Chapter 5 Choosing a Menu Style</b>	<b>65</b>
Menus	66
What is a Menu?	66
Menu Styles	66
Choosing a Menu Style	67
How to Choose a Menu Style	67
<b>Chapter 6 Creating a Menu Manager Menu</b>	<b>69</b>
Menu Manager Menu	70
Menu Manager Menu Illustrated	70
Procedure	70
Defining the Physical Layout	71
Adding Functionality to a Menu	74
Starting a New Transaction	74
Calling a Rule	75
Calling Another Menu	76
<b>Chapter 7 Creating a Screen Manager Menu</b>	<b>77</b>
Screen Manager Menu	78
Predefined Screen Requirements	78
Screen Manager Menu Illustrated	78
Procedure	78
Defining the Physical Layout	79
Adding Functionality to a Menu	83
Starting a New Transaction	83
Calling a Rule	84
Calling Another Menu	85
<b>Chapter 8 Creating a Selection Manager Menu</b>	<b>87</b>
Selection Manager Menu	88
Predefined Screen Requirement	88
Selection Manager Menu Illustrated	88
Procedure	89
Defining the Physical Layout	90
Example	90
Adding Functionality to a Menu	94
Starting a New Transaction	94
Calling a Rule	95
Calling Another Menu	96

<b>Chapter 9 Creating a Standard Session Manager Menu</b>	<b>97</b>
Standard Session Manager Menu	98
Standard Session Manager Menu Illustrated	98
Procedure	98
Defining the Physical Layout	99
Example	99
Adding Functionality to a Menu	103
Starting a New Transaction	103
Calling a Rule	104
Calling Another Menu	105
<b>Chapter 10 Maintaining Menus</b>	<b>107</b>
Copying Menu Definitions	108
Creating Menu Definition Copies	108
Copying Standard Session Manager Menus	108
Editing Menus	109
PF Keys and Primary Commands	109
Deleting a Menu Definition	110
Considerations when Deleting a Menu	110
Deleting a Definition Using the Menu Definer	110
Deleting a Definition Using a Shareable Tool	111
Adding Help Information	112
Adding Help Information	112
SCRIPT Commands	112
Help Screen Restrictions	113
<b>Appendix A Displaying an Application</b>	<b>115</b>
Displaying a Screen	116
Example of the DISPLAY Statement	116
Displaying a Menu	117
Example of the DISPLAY_MENU Tool	117
Using the PROCESS_FCNKEY Tool	118
Examples	118
<b>Appendix B Display Masks</b>	<b>121</b>
Display Masks	122
Numeric Display Masks	122
Date Display Masks	123
<b>Appendix C Sample Screen Definitions for Menus</b>	<b>127</b>

Definition for the WEEKLY\_REPORT Screen ..... 128

    WEEKLY\_REPORT Screen ..... 128

    TITLE Screen Table ..... 129

    SELECTION Screen Table ..... 130

    PFKEY\_SPECS Screen Table ..... 131

    SESSMGR\_MSG Screen Table ..... 132

Definition for the WEEKLY\_REP Screen ..... 133

    WEEKLY\_REP Screen ..... 133

    TITLE\_1 Screen Table ..... 134

    FIELD Screen Table ..... 135

    PRESS Screen Table ..... 136

    PFKEY\_SPECS Screen Table ..... 136

    SESSMGR\_MSG Screen Table ..... 137

Definition for SCR\_EMPLOYEE Menu ..... 138

    Screen 1 ..... 138

    Screen 2 ..... 139

    Screen 3 ..... 140

    Screen 4 ..... 141

**Index ..... 143**



# Preface

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

This manual provides the basic information to define TIBCO Object Service Broker screens, screen tables, and menus.

## Topics

---

- [Related Documentation, page x](#)
- [Typographical Conventions, page xv](#)
- [Connecting with TIBCO Resources, page xviii](#)

## Related Documentation

---

This section lists documentation resources you may find useful.

### TIBCO Object Service Broker Documentation

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

#### Fundamental Information

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

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

## Application Development and Management

The following manuals provide information about application development and management:

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

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

## System Administration on the z/OS Platform

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

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

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

## System Administration on Open Systems

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

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

## External Database Gateways

The following manuals describe external database gateways:

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

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

## Typographical Conventions

The following typographical conventions are used in this manual.

Table 1 General Typographical Conventions

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

Table 1 General Typographical Conventions (Cont'd)




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

Table 2 Syntax Typographical Conventions

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



Table 2 Syntax Typographical Conventions

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

## Connecting with TIBCO Resources

---

### How to Join TIBCOmmunity

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

### How to Access All TIBCO Documentation

You can access TIBCO documentation here:

<http://docs.tibco.com>

### How to Contact TIBCO Support

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

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

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

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

<https://support.tibco.com>

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

## Chapter 1

# Introduction to TIBCO Object Service Broker Screens

This chapter describes what Screens, Menus, and Validation are in the context of TIBCO Object Service Broker.

## Topics

---

- [TIBCO Object Service Broker Screens, page 2](#)
- [TIBCO Object Service Broker Menus, page 3](#)
- [Validation, page 4](#)

## TIBCO Object Service Broker Screens

---

### What is a Screen?

Applications use screens to convey information to and from users. A screen is built from screen tables, which are specialized tables defined through the Screen Table Painter. A screen appears in text mode in a graphical environment. For more information on screens, refer to [Chapter 2, Building a Screen, on page 7](#).

### What is a Screen Table?

A screen table is a type of TIBCO Object Service Broker table used to display data on a screen. The actual data is stored in other types of tables. A screen table is defined to a screen and can be shared with other screens. It has a table type of SCR. For more information on screen tables, refer to [Chapter 3, Defining Screen Tables, on page 25](#).

### Advantages of Using Screen Tables

The TIBCO Object Service Broker process of building screens from smaller components offers the following advantages:

- You can create and modify screens using previously defined components that are already used elsewhere.
- Applications can manipulate screen tables the same way as data tables; occurrences can be retrieved, inserted, and deleted, and values can be assigned to fields.

# TIBCO Object Service Broker Menus

---

## What is a Menu?

A menu is a list of options presented on a screen for selection. A menu can be part of an application; it can be a login screen for a group of users or it can be an individualized login screen. Menus are created through the [DEFINE\\_MENU](#) tool.

## Menu Styles

There are four different menu styles available to choose from when you are defining a menu. The styles you choose depend on your design requirements and intended use. Some styles require preliminary preparation such as predefined screens and applications, while others can be used directly.

The four menu styles, and their corresponding chapter, are:

Menu Manager	<a href="#">Chapter 6, Creating a Menu Manager Menu, on page 69</a>
Screen Manager	<a href="#">Chapter 7, Creating a Screen Manager Menu, on page 77</a>
Selection Manager	<a href="#">Chapter 8, Creating a Selection Manager Menu, on page 87</a>
Standard Session Manager	<a href="#">Chapter 9, Creating a Standard Session Manager Menu, on page 97</a>

## Validation

---

Validation is the check performed on data entered into a field to verify if it conforms to the known characteristics of that field. For example, known characteristics could be semantics, formatting, association, syntax, and so on.

### Phase One Validation

In Phase One validation, data is validated for its semantic data type (logical, date, or numeric). A field containing invalid data is highlighted, the cursor is positioned on the field, and a message appears. Phase One validation of data begins when you press any PF key, except the Refresh key PF24.



All modified screen table occurrences pass through Phase One validation.

### Phase Two Validation

The following process occurs in Phase Two validation:

1. Data is checked to see if any fields containing null values require data.
2. Data is checked to see if values supplied for any fields referencing other tables are valid.
3. Validation rules specified for any of the screen tables are invoked.

Phase Two validation begins when you press any assigned PF key, except the Refresh key PF24, the scroll keys, or the Help key.



All screen table occurrences pass through Phase Two validation.

### Validation Failure

If validation fails in either Phase One or Phase Two, you can do one of the following:

- Press PF12, the Validation Exit key. This raises the exception `VALIDATEFAIL` and you exit the screen display.

If `VALIDATEFAIL` is raised, the data can be accessed under the exception handler `ON VALIDATEFAIL`. The invalid data is reset to null after Phase One

validation. After Phase Two validation, the invalid data is not reset to null; it is available to the exception handler as entered into the screen.

- Type a new value and press one of the assigned PF keys (except for the Refresh key PF24) to initiate the validation process again.

For information on the Validation Exit key (PF12), refer to [Chapter 2, Building a Screen, on page 7](#).

**See Also** *TIBCO Object Service Broker Programming in Rules* for information on exception handling.





## Chapter 2      **Building a Screen**

This chapter describes how to use the Screen Definer to define the overall structure and functionality of a screen.

### Topics

---

- [The Screen Definer, page 8](#)
- [Task A: Verify Screen Identification, page 10](#)
- [Task B: Specify Screen Tables, page 11](#)
- [Task C: Enter PF Key, Scrolling, and Cursor Information, page 15](#)
- [Task D: Specify PF Keys, page 18](#)
- [Deleting a Screen Definition, page 20](#)
- [Maximizing Screen Layout, page 22](#)

## The Screen Definer

---

You use the Screen Definer tool to define the overall structure and functionality of a screen. Using the Screen Definer, you can define the structure of the screen by adding screen tables to the screen definition. You can also specify the PF keys for the screen and the function of the **Scroll** field and cursor.

### Accessing the Screen Definer

You can access the Screen Definer from the workbench by doing one of the following:

- Type a new or existing screen name to the right of the DS define screen option and press Enter. This displays the initial Screen Definer screen.
- Position your cursor to the right of the DS define screen option and press Enter. This displays the Object Manager screen for the Screen Definer. Select an object from this screen to invoke the Screen Definer screen.
- Type DS and a new or existing screen name in the command line.
- Type DS only in the command line and press Enter, and then select a screen name from the Object Manager screen.
- Execute the tool **DRAW** (*screenname*) where *screenname* is the name of a screen.

You can also access the Screen Definer from within the OS object set workbench option. To access the Screen Definer, from within the first screen, position your cursor on the name of a screen object and press PF9.

### Layout

The Screen Definer screen is divided into the following four segments:

- Screen identification
- PF key, scrolling, and cursor information
- Screen table information
- PF keys for the Screen Definer

### Example

The following example illustrates the definition for the EMPLOYEE\_EXPENSE screen:

COMMAND ==>				DEFINE SCREEN: EMPLOYEE_EXPENSE				Unit: ACC			
PFkey Assignments				Scroll Amount Entry				Default Cursor Position			
-----				-----				-----			
Up: 7		Down: 8		Table: COMP_TITLE				Table: EXPENSE_DATA			
Left: 10		Right: 11		Field: SCROLL_AMT				Field: MONTH			
Validation Exit:		12									
Help: 1		Refresh: 24									
SCREEN TABLES:		Origin		Max				Title		Fix	
Name		Row Col		occ Scroll		Validation Rule		Rows		Col	
-----		---		---		-----		---		---	
_ COMP_TITLE		1 1		1 N				0		0	
_ EXPENSE_DATA		5 1		* Y		VALID_NUMBER		5		0	
_ EXPENSE_TOTAL		-5 1		1 N				0		0	
_ FCNKEY_SPECS		-1 1		1 N				0		0	
PFKEYS: 6=PAINT 16=EXCLD 21=DISPLAY 18=SPEC_FCNKEYS 19=SCR_HELP 13=PRINT											

Tasks for Defining a Screen

The following table lists the tasks required to define a screen:

- [Task A: Verify Screen Identification, page 10](#)
- [Task B: Specify Screen Tables, page 11](#)
- [Task C: Enter PF Key, Scrolling, and Cursor Information, page 15](#)
- [Task D: Specify PF Keys, page 18](#)



To preview the layout of a screen, press PF21. If during the screen definition session you save objects in the Screen Table Painter, you cannot display a screen.

## Task A: Verify Screen Identification

### Purpose of this Task

This task is used to uniquely identify the screen and its application or logical unit.

### Screen Identification Segment

The following example illustrates the fields used to identify the screen:

DEFINE SCREEN: EMPLOYEE_EXPENSE	Unit: ACC
---------------------------------	-----------

### The Define Screen and Unit Fields

The information for both the **DEFINE SCREEN** and **Unit** fields are entered by default. You can modify either one of the fields, if necessary.

<b>DEFINE SCREEN</b>	The screen name displayed in the <b>DEFINE SCREEN</b> field is the one you specified when you invoked the Screen Definer. You can type a new name to save the definition of the existing screen under the new name.
<b>Unit</b>	The unit marks the screen as belonging to a particular application or logical unit such as utilities, accounting, or network control. You can modify this as required.

See Also *TIBCO Object Service Broker Shareable Tools* for information on copying TIBCO Object Service Broker objects.

# Task B: Specify Screen Tables

## Purpose of this Task

This task is used to specify the attributes and position of each screen table on the screen. There are two types of attributes you can specify:

- Required
- Optional

## Screen Table Segment

The following example illustrates the fields used to specify screen tables. To view the **ShowPartial Occs** and **Empty Occs** fields to the right, press PF11.

SCREEN TABLES: Name	Origin Row Col	Max occ	Scroll	Validation Rule	Title Rows	Fix Col	Last Row Col
-----	---	---	-	-----	--	---	---
_ COMP_TITLE	1 1	1	N	VALID_NUMBER	0	0	0 0
_ EXPENSE_DATA	5 1	*	Y		5	0	0 0
_ EXPENSE_TOTAL	-5 1	1	N		0	0	0 0
_ FCNKEY_SPECS	-1 1	1	N		0	0	0 0

## Defining Required Screen Table Attributes

The following fields are used to define the required screen table attributes. For each field, press PF1 for valid values.

Name	This field indicates the name of the screen table. Since screen tables are shareable, you can type the name of an existing screen table or create a new screen table. The maximum number of screen tables that can be specified for a screen is 25.
Origin Row	<p>This field identifies the first row of the screen table. The screen length used for validation is the number of rows on your screen. Negative numbers indicate the distance from the bottom of the screen to the row. Positive numbers indicate the distance from the top of the screen to the row.</p> <p>For information on maximizing screen layout, refer to <a href="#">Maximizing Screen Layout on page 22</a>.</p>

<b>Origin Col</b>	<p>This field identifies the first column of the screen table. The screen width used for validation is the number of columns on your screen. The maximum width of a screen table is 254 characters. Negative numbers indicate the distance from the right side of the screen to the column.</p> <p>For information on maximizing screen layout, refer to <a href="#">Maximizing Screen Layout on page 22</a>.</p>
<b>Max occ</b>	<p>This field specifies the maximum number of row occurrences that can be inserted into the screen table. This is not the maximum number of occurrences that can appear at one time. An asterisk (*) indicates an unlimited number of occurrences.</p>
<b>Scroll</b>	<p>This field determines whether the screen table is scrollable both horizontally and vertically.</p>

## Defining Optional Screen Table Attributes

The following fields are used to define optional screen table attributes. To view additional fields, press PF11. For valid values, press PF1.

<b>Validation Rule</b>	<p>This field indicates the name of the validation rule to be executed during a screen display. The rule ensures that changes made to the data in a screen table satisfy criteria specified in the rule. The validation rule is executed once per display and is also executed after any existing reference check is performed on a field.</p> <p>Refer to <a href="#">Chapter 3, Defining Screen Tables, on page 25</a> for more information on reference checking and <i>TIBCO Object Service Broker Managing Data</i> for more information on validation rules.</p>
<b>Title Rows</b>	<p>This field specifies the number of lines to be treated as title rows in scrollable screen tables. The first &lt;nn&gt; lines of the screen table remain on the screen when scrolling up or down.</p>
<b>Fix Col</b>	<p>This field is used for scrollable screen tables. The <b>Fix Col</b> field identifies the number of columns in a screen table that are to remain on the physical screen when the screen is scrolled left or right. These columns are typically grouped together to form a column of data.</p> <p>Literal text created in the Screen Table Painter is split based on these fixed columns. After you save the screen definition, a message could appear on the workbench to inform you of overlap conditions. Press PF2 to display a list of screen fields that span the fixed column boundary.</p>
<b>Last Row</b>	<p>This field specifies the last row the screen table occupies on the screen, so that you can set the height of the screen table on the screen. If it is set to 0 or null, the screen table fills all the rows until another screen table starts. If you set this attribute, you must also set the last column attribute.</p> <p>For information on maximizing screen layout, refer to <a href="#">Maximizing Screen Layout on page 22</a>.</p>

<b>Last Col</b>	<p>This field specifies the last column the screen table occupies on the screen, so that you can specify the right edge of the screen table on the screen. If it is set to 0 or null, the screen table occupies the entire remainder of columns. If you want to have screen tables side by side on the screen, you must use this attribute to set the right edges of the screen tables. If you set this attribute, you must also set the last row attribute.</p> <p>For information on maximizing screen layout, refer to <a href="#">Maximizing Screen Layout on page 22</a>.</p>
<b>ShowPartial Occs</b>	<p>This field determines whether part of an occurrence should appear on the screen if the occurrence spans multiple rows (or lines) and cannot fit on one screen.</p>
<b>Empty Occs</b>	<p>This field controls the number of empty occurrences of a screen table that appear on the user's terminal. The <b>Empty Occs</b> specification provides a quick method for developers to display screen occurrences for data entry applications automatically, using either the default null specification or a value greater than zero. A value of zero disables this feature and forces the application rules to control the screen display content through INSERTs of the screen table prior to the screen DISPLAY.</p> <p>The title literals always appear even if this field is set to 0.</p>



For a screen table to appear, it must begin on the displayable screen image. For example, if a screen table begins in column 81 and it appears on a screen that is only 80 columns wide, it is not visible even by scrolling. The number of displayed columns in the screen table is the maximum amount scrolled when scrolling the table left or right.

### Excluding Screen Tables

If after adding a screen table, you want to exclude or remove it from the current screen definition, complete the following tasks:

1. Position the cursor on the screen table you want to exclude.
2. Press PF16.

The screen table is excluded from the definition; it is not deleted from the system. You remain in the screen you are defining.



# Task C: Enter PF Key, Scrolling, and Cursor Information

## Purpose of this Task

This task is used to:

- Assign PF key values
- Define the scroll amount entry
- Define the default cursor position

## PF Key, Scrolling, and Cursor Information Segment

The following example illustrates the fields used to enter PF key, scrolling, and cursor information:

PFkey Assignments	Scroll Amount Entry	Default Cursor Position
-----	-----	-----
Up: 7      Down: 8	Table: COMP_TITLE	Table: EXPENSE_DATA
Left: 10    Right: 11	Field: SCROLL_AMT	Field: MONTH
Validation Exit: 12		
Help: 1    Refresh: 24		

## Assign PF Key Values

You can specify which PF keys are used to scroll screen tables, exit, and access help when the user displays a screen.

## Exceptions

You can optionally re-assign all the PF key values except for the Refresh key. TIBCO Object Service Broker automatically assigns PF24 as the key to refresh your screen to its state after the last command was issued (works for primary and line commands).

## Changes

To change the default for one of the PF keys, specify an integer denoting a previously undefined PF key. You can disable a function by typing zero into the field.

Validation Exit

The Validation Exit key is used to escape from situations when a validation failure occurred. Normally all PF keys are ignored until the screen passes validation; however, the Validation Exit key is processed during a validation failure. The result of using the Validation Exit key during a validation failure is that the exception `VALIDATEFAIL` is raised. For more information on the `VALIDATEFAIL` exception, refer to *TIBCO Object Service Broker Programming in Rules*.

Using the Validation Exit key on a valid screen has no effect and normal PF key processing takes place. This means you can assign another function to the Validation Exit key using the `FCNKEYS` table.

Define the Scroll Amount Entry

You can define one of the fields of a screen table to work with the scroll keys, if required. Your user can then use the **Scroll** field (displayed in the upper right corner of the screen) to specify a valid scroll amount when using your screen.

The following fields are used to define the scroll amount entry. Before you define these fields, you must define the field to work with the scroll keys using the Screen Table Painter. For more information on the Screen Table Painter, refer to [Chapter 3, Defining Screen Tables, on page 25](#).

Table	Enter the name of the screen table. This must be an existing screen table in the screen you are defining and the screen table must contain the <b>scroll</b> field.
Field	Enter the name of the field. This must be an existing field in the screen table named in the <b>Table</b> field. This is where the user specifies the scroll amount.



- When defining a scroll amount entry:
- The scroll amount field must be in a single occurrence screen table or in the title area of a multi-occurrence screen table.
  - The scroll amount field should have a syntax of C.



If you defined a scroll amount field for a screen in the scroll amount entry segment from the Screen Definer, make sure you initialize it with a value before you display the screen.

## Define the Default Cursor Position

When the user first displays a screen, the cursor is located in the first position of the default field. If there is no default field, the cursor is located in the first unprotected field starting from left to right then top to bottom. You can define a default cursor position, if required.

The following fields are used to define the default cursor position. Before you define these fields, you must define the field for the default cursor position using the Screen Table Painter. For more information on the Screen Table Painter, refer to [Chapter 3, Defining Screen Tables, on page 25](#).

<b>Table</b>	Enter the name of the screen table. This must be an existing screen table in the screen you are defining and the screen must contain the default field.
<b>Field</b>	Enter the name of the field using the Screen Table Painter. This must be an existing field in the screen table named in the <b>Table</b> field. This field is where the cursor is to appear when the user displays the screen.

## Task D: Specify PF Keys

### Assigning PF Keys

PF keys, also referred to as function keys, can be specified as follows:

1. Position your cursor on the screen table you defined for your PF keys.
2. Press PF18.

The @TMP\_FCNKEYS table appears. You can use this table to define the function keys that appear on the FCNKEY line at the bottom of the screen you are defining.

3. Specify the attributes.

Refer to [Specifying Function Key Attributes](#) below for information on specifying attributes for function keys.

4. Press PF3 to save and exit the @TMP\_FCNKEYS table.

You are returned to your screen definition.

### Specifying Function Key Attributes

To specify function key attributes, make the following entries:

PF_KEY	Type a valid PF key.
NAME	Type the name of the PF key.
COMMAND	Type the name of the primary command associated with the PF key, if required. You must define a primary command field in one of your screen tables to use this entry.
ROUTINE	Type the name of the rule. The rule is invoked when the PF key is used. The rule contains code for the function associated with the PF key.
DESCRIPTION	Provide a longer description of the PF key. This is used in the help for your screen.
MSG_INCLUDE	Indicate whether the PF key is listed on the message line.

---

<b>MSG_ORDER</b>	Place the PF key in a position on the message line. Type a positive integer or leave it blank. The default is zero.
------------------	---

---



The [PROCESS\\_FCNKEY](#) tool can be used to process the PF keys that you define in this manner. For more information on [PROCESS\\_FCNKEY](#), refer to *TIBCO Object Service Broker Shareable Tools*.

## @TMP\_FCNKEYS Table Example

The following example illustrates the @TMP\_FCNKEYS table. To view additional fields, press PF11.

---

EDITING TABLE : @TMP\_FCNKEYS  
COMMAND ==>

PF_KEY	NAME	COMMAND	ROUTINE	SCROLL: P
PF12	EXIT		EXIT_DISPLAY	
PF13	PRINT		PRINT_SCREEN	
PF3	SAVE		SAVE_INFO	
PF9	TOTAL		TOTAL_EXP	

PFKEYS: 4=INSERT 16=DELETE 5=FIND NEXT 6=CHANGE NEXT 18=EXCLUDE 3=SAVE

---

## Deleting a Screen Definition

There are a number of methods that you can use to delete a screen definition. Refer to the following table to determine the appropriate method to use:

If the screen...	Use...
Was promoted to a target system	Promotion system for that location.
Was not promoted and is on your local node	Screen Definer or <a href="#">DELETE_DEFN</a> .
Was not promoted and is on a node remote to the node where you are presently working	<a href="#">DELETE_DEFN</a> .

Deleting a screen definition does not delete the screen tables included in the screen definition.

### Considerations when Deleting a Screen

If a screen was promoted to another (target) system, you must submit a change request through the Promotion system (of the source system) to have the deletion extended to the target system. If you do not issue a change request to delete the definition, the following occurs:

- The screen exists on the target system and no rights are associated with it on the source system.
- If a new screen with the same name is created on the source system, the screen cannot be promoted to the target system because a screen with the same name already exists there.

### Deleting a Definition Using the Screen Definer

When you are within an existing definition in the Screen Definer you can delete a definition by doing one of the following:

- Press PF22.
- Use the **DELETE** command.

In either case you are prompted to confirm the deletion.

## Deleting a Definition Using a Shareable Tool

The shareable tool `DELETE_DEFN` is available to you to delete existing definitions. Using this tool, if the correct security access is set up, you can delete definitions across TIBCO Object Service Broker nodes or within your local node. `DELETE_DEFN` is called from within a rule.

### Example Rule

The following rule calls the `DELETE_DEFN` tool to delete the definition of the DEPARTMENTS screen from Node A. Since a screen definition is being deleted, a value is not required for the argument *library*.

```
DELETE_SCREENEFN;  
-----  
CALL DELETE_DEFN('SCREEN', 'DEPARTMENTS', '', ' ', 'NODEA', | 1  
                  'Y');                                     |  
-----
```

- See Also
- TIBCO Object Service Broker Managing Deployment* for information about change requests.
  - TIBCO Object Service Broker Shareable Tools* for information about the tools.

## Maximizing Screen Layout

The **Origin Row** and **Col** fields combined with the **Last Row** and **Col** fields can be used to define the layout of the screen. Use negative values for the origin row of the bottom of the screen table to ensure they appear at the bottom of the screen even for different terminal sizes. If you want to have screen tables side by side, you must use the Last Row and Col attributes to specify the bottom right corner of the left screen tables.

### Screen Table Definitions

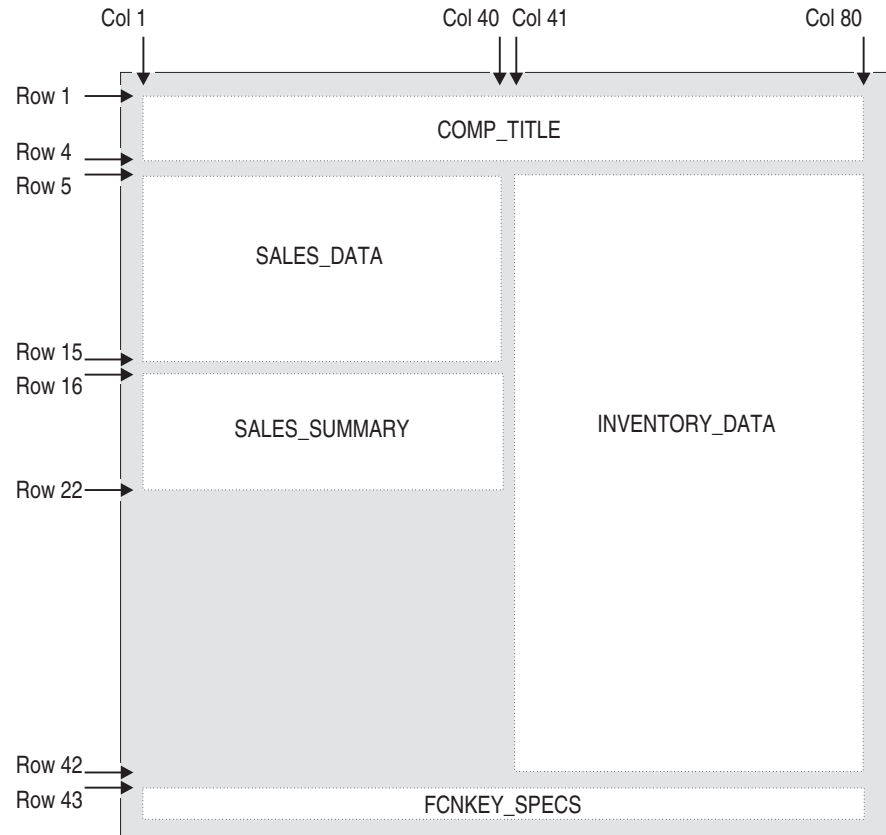
The following example illustrates screen table definitions that are used for reference in the following two diagrams.

COMMAND ==>										DEFINE SCREEN: SALES_INFO										Unit: USR40									
PFkey Assignments										Scroll Amount Entry										Default Cursor Position									
-----										-----										-----									
Up: 7					Down: 8					Table:					Table:														
Left: 10					Right: 11					Field:					Field:														
Validation Exit: 12																													
Help: 1 Refresh: 24																													
										</																			



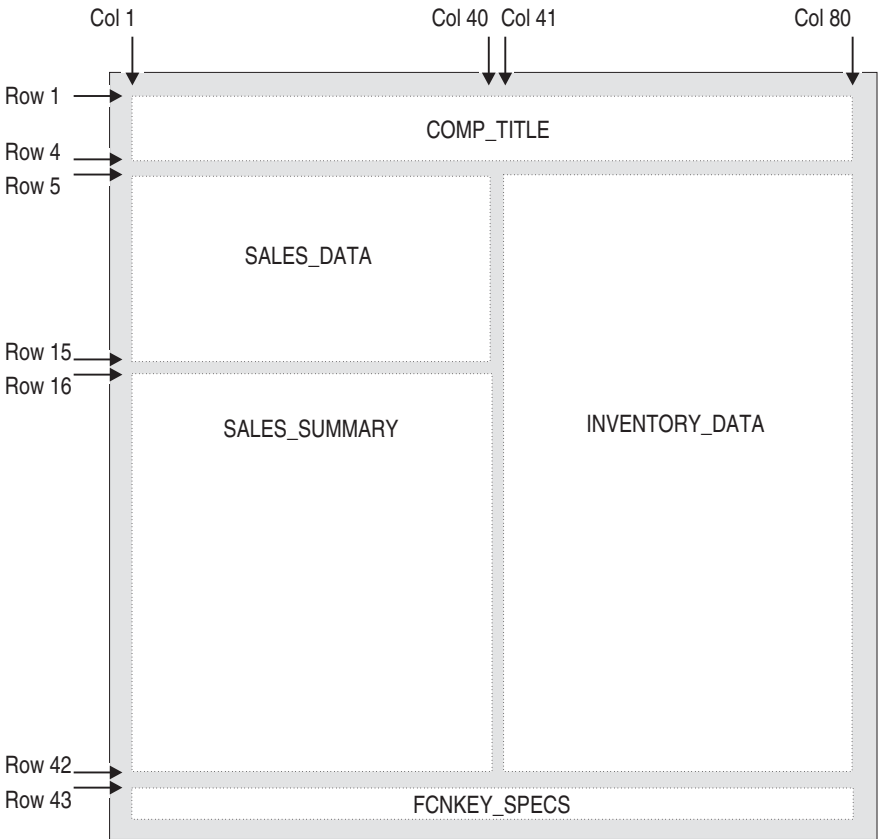
## Last Row Attribute Set to 22

Notice that in the following diagram there is wasted space below the SALES\_SUMMARY screen table. This can be avoided by using negative numbers for the Last Row attribute. If the Last Row attribute for the SALES\_SUMMARY screen table were set to -2 instead of 22, the screen table would expand to fill all the space in the 80x44 screen.



### Last Row Attribute Set to -2

This diagram illustrates the layout of the screen when the Last Row attribute for the screen table SALES\_SUMMARY is set to -2 instead of 22:



## Chapter 3      **Defining Screen Tables**

This chapter defines Screen Tables and Screen Fields and describes how to use the Screen Table Painter.

### Topics

---

- [Screen Tables and Screen Fields, page 26](#)
- [The Screen Table Painter, page 27](#)
- [Using the Screen Table Painter, page 28](#)
- [Adding Literal Text, page 31](#)
- [Defining Data Fields, page 34](#)
- [Using Predefined Fields, page 36](#)
- [Using Fields with Default Attributes, page 40](#)
- [Creating New Fields, page 41](#)
- [Controlling Presentation, page 44](#)
- [Expanding Definitions, page 47](#)
- [Editing Screen Tables and Screen Fields, page 49](#)

## Screen Tables and Screen Fields

---

### Screen Tables

Screen tables are virtual tables used to display the user interface of an application. Each screen is made up of one or more screen tables and these screen tables can be shared among screens.

### Table Type

Screen tables have a table type of SCR and, unlike most table types, they are defined using the Screen Table Painter and not the Table Definer. Screen tables are not used to store data; they are used to access and display data.

### Screen Fields

Screen fields are the fields used to display data within a screen table. You can define data and presentation attributes to screen fields. As with any other TIBCO Object Service Broker table, each field must be unique within a screen table, although the fields can have the same name as fields within other tables. This makes it easy to share data.

### Screen Field Limit

When defining screen fields, if the number of fields exceeds 150, a warning message appears on the Screen Definer screen upon return from the Screen Table Painter. You can change the limit of screen fields in your configuration. For more information on changing your configuration, refer to *TIBCO Object Service Broker for z/OS Installing and Operating* or *TIBCO Object Service Broker for Open Systems Installing and Operating*.

## The Screen Table Painter

---

You use the Screen Table Painter tool to define the specialized tables and fields that make up a screen. In the Screen Table Painter, you can specify literals that appear on the screen and their attributes, and define screen fields and their attributes.

### Invoking the Screen Table Painter

To invoke the Screen Table Painter from the Screen Definer, complete the following tasks:

1. In the Screen Definer, specify the screen tables that make up your screen.  
Refer to [Chapter 2, Building a Screen, on page 7](#) for information on specifying screen tables for a screen.
2. Place the cursor anywhere on the row containing the screen table name that you are defining.
3. Press PF6.

If the screen table already exists, the Screen Table Painter displays its definition; otherwise, the Screen Table Painter displays an empty template for you to complete.

# Using the Screen Table Painter

## Layout

The Screen Table Painter screen is divided into the following three areas:

- Image area
- Field definition area
- Selected PF keys for the Screen Table Painter

## Screen Table Painter Screen

The following example illustrates the Screen Table Painter for the screen table EXPENSE\_DATA:

SCREEN PAINTER COMMAND ==>Scroll: P  
.....1.....2.....3.....4.....5.....6.....7.....  
-999999 Expenses for Department #-99  
Manager #-99999

EMPLOYEE NAMEEMPLOYEE#EXPENSE  
-AAAAAAAAAAAAAAAAAAAAA-99999-99999.99

Table: EXPENSE\_DATAUnit: USR40

ROW	COL	FIELD NAME	Type	Syn	Len	Dec	Just	Fill	Prot	Show	Rqd	Hi	Skip	Null
1	16	MONTH	D	B	8	0	L	.	Y	Y	N	N	Y	Y
1	49	DEPTNO	I	P	2	0	L		Y	Y	N	N	Y	Y
2	49	MANAGER_NUM	I	P	5	0	L		Y	Y	N	N	Y	Y
6	1	LNAME	S	C	22	0	C		Y	Y	N	N	Y	Y
6	32	EMPNO	I	P	5	0	L		Y	Y	N	N	Y	Y
6	45	MONTHLY_EXP	Q	P	8	2	R	_	N	Y	N	N	Y	Y

PFKEYS: 6=+FLD 18=-FLD 4=+LINE 5=CUT 19=FLD\_HELP 17=PASTE 16=-LINE 13=PRINT

## Image Area

The image area displays the appearance of the screen table. It is composed of the following elements: a ruler, literal text, and screen fields. In the image area, you can do the following:

- Type in and position literal text up to 254 characters per row
- Create and position screen fields
- Copy and append fields from other tables
- Cut and paste fields
- Add and delete fields
- Copy, add, and delete lines (literal text and screen fields)
- Change the syntax of fields
- Change the length of fields

The image area scrolls both horizontally and vertically and a scroll amount field is also available.

## Field Definition Area

The field definition area displays attributes for each of the screen fields associated with a specific screen table. It contains two sections that you can edit:

- Header information
- Field information

### Header Information

The header information section, which is non-scrollable, contains default values for the screen table name and the unit to which the screen table belongs. You can change either of these values.

To use this definition as a template for a new definition, change the screen table name. The screen fields are associated with this new screen table. For more information on copying TIBCO Object Service Broker objects, refer to *TIBCO Object Service Broker Shareable Tools*.

### Field Information

The field information section, which is scrollable, contains the attributes for each of the fields that you define. You can enter a definition directly into this area.

If you add fields using the image area, default information is added for you in the field definition section. You can modify these default values as required.

## Available PF Keys for the Screen Table Painter

The PF key line displays some of the most commonly used PF keys for the Screen Table Painter. Not all the most commonly used PF keys appear on this line. For information on all available PF keys, press PF1 for a listing and descriptions. For detailed information on the standard PF keys, refer to *TIBCO Object Service Broker Getting Started*.

## Defining Screen Tables

There are two components to defining a screen table:

- Adding literal text
- Defining data fields

These two components are discussed in the following sections.



## Adding Literal Text

---

### Definition

The text used in titles and footers, field labels, and on-screen instructions is referred to as literal text. During the display of a screen, this text is protected (the user cannot change the text) and remains constant.

### Example

The following example illustrates literal text on a screen:

---

Expenses for Department # Manager #		
EMPLOYEE NAME	EMPLOYEE#	EXPENSE

---

### Creating Literal Text

To create literal text, type the text that you want to use into the image area, exactly as you want it to appear on your screen. After creating literal text, you can do the following:

- Set display attributes
- Expand literal text

### Considerations

Note the following about literal text:

- Literal text cannot start in the first column of your screen.
- You must have one blank space between literal text and its adjacent data fields.

## Setting Display Attributes

You can control the presentation of literals by setting the display attributes. To set the display attributes, complete the following tasks:

1. Press PF9 from anywhere on the Screen Table Painter screen.

You do not have to be positioned on the literal text. The LITERAL ATTRIBUTES screen appears. This screen shows sample literals and how the literal text looks with the attributes set.

2. Specify the attributes.

To see any changes to attributes, press Enter and the changes are shown.

## Display Attributes for Literals

There are six attributes you can set:

- Highlight
- Blink
- Reverse video
- Underline
- Foreground color
- Background color

If you set any of these attributes, they apply to all literals in the screen table. Different screen tables can have different literal attributes. Therefore, if you want to have different attributes, design your screen accordingly by grouping fields that have both content and attributes in common.

## Literal Attributes Screen

The following example illustrates the LITERAL ATTRIBUTES screen for the screen table EXPENSE\_DATA.

---

LITERAL ATTRIBUTES for screentable EXPENSE\_DATA

-----  
This line shows how the literals will look.  
-----

Attributes	Colour
Highlight : Y	Foreground :
Blink : N	Background :
Reverse : N	
Underline : N	

PFKEYS: 3=SAVE 12=EXIT

---

## Defining Data Fields

---

### Methods for Defining Data Fields

There are three methods for defining fields that contain data:

- Using predefined fields
- Using fields with default attributes
- Creating new fields

The following sections describe the benefits of using each of these for creating screen fields.

### Benefits of Using Predefined Fields

There are two types of predefined fields you can use when creating your screen field definitions:

- Table field definitions
- Global fields

The following sections describe the benefits for using both methods. For information on how to use predefined fields, refer to [Using Predefined Fields on page 36](#).

#### Table Field Definitions

You can use fields and their definitions from a predefined table in your screen table definition to promote the shareability of data definitions. The use of definitions for screen fields simplifies coding and the assignment of data. If the screen fields match the table fields, you do not have to worry about matching the appropriate fields or leaving fields out.

#### Global Fields

Global fields are another type of predefined field that you can use in your screen table definition. Global fields are fields defined by your system administrator to be used system wide for all table types (TDS, RPT, SCR, and so on). Global fields are used to create standard definitions across the database. Global fields standardize the position, name, and how your fields use data across your screens. You can then customize the presentation of each field when you copy the field into your definition or at a later time.

Global fields are found in the Global Field dictionary. Consult your system administrator for more information.

## Benefits of Adding Fields with Default Attributes

You can create screen fields that contain default attributes. Using default attributes you can achieve the functionality of the field and customize the presentation of the field at a later time. Using screen fields with default attributes is useful if you want to use the field for testing purposes and are not concerned with the presentation of the field.

For information on creating screen fields with default attributes, refer to [Using Fields with Default Attributes on page 40](#).

## Benefits of Creating New Fields

Creating new fields gives you total control over the data definition and the presentation attributes. You can create screen fields to your specifications. For information on creating new fields, refer to [Creating New Fields on page 41](#).

## Using Predefined Fields

You can use predefined fields to create definitions for your screen fields, unless these fields have a syntax of RD (raw data) or UN (Unicode). There are two types of predefined fields:

- Table field definitions
- Global fields

### Using Table Field Definitions

You can use a field(s) and its definition from a predefined table to simplify the assignment of data and promote the shareability of data definitions. To use a field in a table, use one of the following primary commands:

- **COPY**
- **APPEND**

### Copying Fields

To copy a field(s) and its definition from a predefined table, use the **COPY** command. The field is copied into the current screen table (the image area is cleared for you when you invoke the command). If there is not sufficient room, only the fields that fit in the screen table are copied. Specify one of the following:

<i>Tablename</i>	A list of fields appears. Place an alphanumeric character, to specify order, in the line command field beside the fields you want to select. They are re-ordered in the order (0-9, A-Z), when you press PF3>.
<i>Tablename.fieldname</i>	The named field is copied into the current screen table.
<i>Tablename.*</i>	From this table, all fields that do not have a syntax of RD (raw data) or UN (Unicode) are copied into the current screen table.

You can also specify the option NONAMES. If the NONAMES option is not used, the field name is copied as a literal with the field added beside it. If the NONAMES option is specified, the field name is not copied as a literal.

## Appending Fields

To append a field(s) and its definition from a named table into the current screen table, use the **APPEND** command. The field is added at the line below the cursor position, if there is sufficient room. If there is insufficient room, only the fields that fit in the screen table are appended. If the cursor is not positioned in the image area, the fields are added after the existing fields. Specify one of the following:

<i>Tablename</i>	A list of fields appears. Place an alphanumeric character, to specify order, in the line command field beside the fields you want to select. They are re-ordered in the order (0-9, A-Z), when you press PF3.
<i>Tablename.fieldname</i>	The named field is appended into the current screen table.
<i>Tablename.*</i>	From this table, all fields that do not have a syntax of RD (raw data) or UN (Unicode) are appended into the current screen table.

You can also specify the option NONAMES. If the NONAMES option is not used, the field name is copied as a literal with the field added beside it. If the NONAMES option is specified, the field name is not copied as a literal.

### Using Global Fields

Global fields standardize the position, name, and how your fields use data across your screens. They are used to create standard definitions across the database. The following example illustrates the Global Field Dictionary screen:

Globalfields		Scroll: P	
COMMAND ==>		Select All: N	
Location:		Deselect All: N	
		Show selection specs: Y	
===== Selection Specification =====			
Selection: NAME LIKE '*'			
AND	Op	Value	
	-----	-----	
	NAME		
	BUSINESSNAME		
	UNIT		
	CREATED		
	AUTHOR		
=====			
Name	Businessname	Unit	Created Aut
-----	-----	-----	-----
_ @ACCESSTYPE	ACCESS FOR IMS, IDMS, ...	XYZ	1987-01-01 AMD
_ DATE	CURRENT DATE	USR40	1995-03-01 USR
_ DEPTNO	DEPARTMENT NUMBER		1992-01-22 MGR
S MANAGER	MANAGER NAME	EMP	1989-03-01 WEA
_ TIME	CURRENT TIME		
S USERID	USERID OF EMPLOYEE		
PFKEYS: ENTER=UPDATE 3=SAVE 12=CANCEL			

### Selecting a Global Field

To select a global field(s) to copy from the Global Field Dictionary, complete the following tasks:

1. Press PF14.  
  
This displays a listing of the Global Field dictionary excluding fields that have a syntax of RD (raw data) or UN (Unicode).
2. To narrow your selection list, beside the appropriate selections, specify an operator in the Op field and appropriate values in the Value field.  
  
You can use the middle section of the screen to narrow down the selection list by using specified selection criteria. The list of fields appears in the lower portion of the screen. You can use more than one type of selection criteria. For a list of valid values for each of these fields, position your cursor on the field and press PF1.
3. Type S in the line command field of the fields you want to copy.



4. Press PF3 to save or copy.

The Screen Table Painter screen appears with the global fields appended to the screen table, one per line in the order in which they are stored in the table. The display lengths of the global fields are used as the lengths of the screen fields.

### Example

The following example illustrates the **MANAGER** and **USERID** fields added to your screen table definition from the Global Field Dictionary:

---

```

SCREEN PAINTER COMMAND ==>                                Scroll: P
....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....
      -999999 Expenses for Department #-99
                                Manager #-99999

      EMPLOYEE NAME      EMPLOYEE#      EXPENSE

-AAAAAAAAAAAAAAAAAAAAA      -99999      -99999.99
-AAAAAAAAAAAAAAAAAAAAA
-AAAAAAAAA

Table: EXPENSE_DATA      Unit: USR40
ROW COL      FIELD NAME      Type Syn Len Dec Just Fill Prot Show Rqd Hi Skip Null
---
1  16  MONTH      D   B   8   0   L   .   Y   Y   N   N   Y   Y
1  49  DEPTNO      I   P   2   0   L   Y   Y   N   N   Y   Y
2  49  MANAGER_NUM  I   P   5   0   L   Y   Y   N   N   Y   Y
6  1   LNAME      S   C  22   0   C   Y   Y   N   N   Y   Y
6  32  EMPNO      I   P   5   0   L   Y   Y   N   N   Y   Y
6  45  MONTHLY_EXP  Q   P   8   2   R   _   N   Y   N   N   Y   Y
7  1   MANAGER      I   C  16   0   L   N   Y   N   N   Y   Y
8  1   USERID      I   C   8   0   L   N   Y   N   N   Y   Y
PFKEYS: 6=+FLD 18=-FLD 4=+LINE 5=CUT 19=FLD_HELP 17=PASTE 16=-LINE 13=PRINT
Copied 2 field(s) from the field dictionary

```

---

## Using Fields with Default Attributes

---

You can create fields for your screen tables that contain default attributes. You can then use the default functionality of the field. You can customize the fields at a later time. This is useful if you want to use the field for testing purposes and are not concerned with the presentation of the field.

### Creating Screen Fields with Predefined Attributes

To create screen fields with predefined attributes, complete the following tasks:

1. Place the cursor one column to the left of where you want the data in the field to begin in the image area.
2. Press PF6.

If there is enough room, a new field is added to the right of where the cursor is positioned.

A marker character, that is, logical not sign (¬) or caret (^), and a sample data element (A or 9) appear. The data element indicates the start of the data in the field. A definition containing default information automatically appears in the field definition area.

## Creating New Fields

---

When you create a screen table you must specify certain fields to contain the data. The other attributes affect the presentation of the field and can be defined at any time. Refer to [Controlling Presentation on page 44](#) for more information on presentation attributes.

### Data Fields

The following fields are required for you to create a screen field to contain the data. For valid values, press PF1.

<b>ROW</b>	<p>Enter a value that determines in which row of the screen table the field appears. To change the row that a screen field is in, use the cut and paste functions. Changes in the row position are reflected automatically with a new value in this field.</p> <p>You can define one or more fields to row 0 and column 0. This creates a non-display field that you can use to pass values to by using rules.</p>
<b>COL</b>	<p>Enter a value that determines in which column of the screen table the field appears. To change the column that a screen field is in, use the cut and paste functions. Changes in the column position are reflected automatically with a new value in this field.</p> <p>You can define one or more fields to row 0 and column 0. This creates a non-display field that you can use to pass values to by using rules.</p>
<b>FIELD NAME</b>	<p>Type a name in this field to uniquely identify the screen field within the screen table. Rules use this name to read data from or write data to the field. Screen fields can be given the same names as fields in other tables; if you are loading data from another table, giving the fields the same name simplifies the process.</p>
<b>Syn</b>	<p>Specify the syntax for the field. The syntax determines how the data is stored. For more information on syntax, refer to <i>TIBCO Object Service Broker Programming in Rules</i>.</p>



- Use non-display fields (**ROW 0, COL 0**) to store information that should not be visible on the screen. For example, you can store information that is should not be released at the present time as it is to be used for future development.
- If the information on the screen comes directly from a MetaStor table, use the same field names for the screen tables as those in the MetaStor. Using the same field names makes the assignment of data to the screen field easier.

## Determining How the Data is Presented

The following fields determine how the data is presented in the screen field. For valid values, press PF1.

<b>Len</b>	Specify a length for the field. This value determines the displayed length of the field allotted to a value of the field when it appears, for example, the maximum number of character positions, including the decimal point, mantissa, and so on. This is done by changing the value of the <b>Len</b> field or by adding characters (A or 9) to the field in the image area.
------------	---



When defining field lengths, the display size does not always correspond to the stored dictionary length. For example, if a field is defined as 10, 2, the underlying field is defined as length 5 and decimal 2. Of those 5 bytes, 9 nibbles are for digits and 1 nibble is for the sign.

<b>Dec</b>	Indicate if you want the field to contain decimals by providing the number of digits to appear to the right of the decimal point. You can also add a decimal point to the field in the image area.
<b>Just</b>	Determines how the data in the field is to be justified. Data can be right, left, or center justified.
<b>Hi</b>	Type Y if you want the field to be highlighted.

## Determining How a Field Operates

The following fields determine how a field operates in the screen table. For valid values, press PF1.

<b>Prot</b>	Type Y if you want to protect a field and use it only for display purposes. If the field is not protected (N), the user can enter data in the field. A screen field cannot be both required and protected.
<b>Show</b>	Type N if you do not want the contents of the field to be shown. An example of this could be a field used for a password.
<b>Rqd</b>	Type Y if you want the user to be required to provide a value in the field. A field requires data only if an occurrence exists to validate. Phase Two validation ensures you supply this data. A screen field cannot be both required and protected.
<b>Skip</b>	Specify Y if you want the cursor to automatically skip to the next field when the last character of the field is entered.
<b>Null</b>	Specify Y if you want the unused portion of the field to be filled with nulls. If the unused portion is filled with nulls, insertions are allowed as nulls can be overwritten. If the unused portion is filled with blanks, no insertions can be made into the field as blanks are characters.

## Controlling Presentation

After creating new fields by specifying data attributes, you can determine the presentation of the field. This can be done when you first define the field or at a later time. Refer to [Creating New Fields on page 41](#) for more information on creating fields with only data attributes.

### Determining Uses for a Field

This section discusses the fields used to determine how a field can be used. For valid values, press PF1.

<b>Typ</b>	Enter the semantic data type of the field. The semantic data type determines how the data in the field is used. For more information on semantic data types, refer to <i>TIBCO Object Service Broker Programming in Rules</i> .
<b>Reference</b>	Specify the name of the reference table. A reference table is used to validate data after the user inserts or modifies it in the field. If the added or modified value does not exist as a primary key value in the referenced table, validation fails. The referenced table cannot be parameterized.
<b>AutoPrompt</b>	Type Y if you want the user to be automatically prompted for selection values when they enter an invalid value or a null value to a required field. This specification is meaningful only for a field that references another table. No prompting is allowed if the field is specified as protected.
<b>UserPrompt</b>	Type Y in the <b>UserPrompt</b> field if you want to supply user prompting for selection values for the field that works in conjunction with the Help PF key. No prompting is allowed if the field is specified as protected.

---

<b>Scrollindicator</b>	<p>Specify if the field is used to indicate in which direction scrolling is available on the screen. If you specify Y in the <b>Scrollindicator</b> field, the field must have these attributes defined as follows:</p> <ul style="list-style-type: none"><li>• Type S</li><li>• Syntax C</li><li>• Length 7</li></ul> <p>The scroll indicator field must be in either the fixed column portion or the non-scrolling header rows of the screen table so that it is always visible on the displayed screen.</p> <p>When the user displays the screen, this field is filled with some combination of the characters less than (&lt;), plus (+), minus (-), and greater than (&gt;). These characters are automatically updated each time the user scrolls. Press PF1 for an explanation of the direction each of the characters indicate. You can define only one scroll indicator field per screen table and you cannot reference this field in a rule.</p>
------------------------	--

---

## Determining Appearance

This section discusses the fields used to determine the appearance of the screen field. For valid values, press PF1.

---

<b>Fill</b>	<p>Specify the character to use as a filler when there is no data for the field or when the data does not completely fill the field. When you access the field through rules, non-blank leading and trailing fill characters are removed.</p>
<b>Display Mask</b>	<p>For a screen field of semantic data type D or syntax P or B, specify a display mask in this field. A display mask determines how a field is formatted. For more information on display masks, refer to <a href="#">Appendix B, Display Masks, on page 121</a>.</p> <p>When accessing fields using a rule, display mask characters are not considered part of the field, therefore, valid lengths are increased and can be anywhere from 1 to 78.</p>

---

<b>Foreground Colour</b>	Specify the foreground color for the field. If your terminal does not support this attribute, it is ignored.
<b>Background Colour</b>	Specify the background color for the field. Background colors are not supported for 3270 terminals. If your terminal does not support this attribute, it is ignored.
<b>Blink</b>	Type Y if you want the field to blink when displayed.
<b>Reverse</b>	Type Y if you want the field to appear in reverse video (the background and foreground colors are reversed).
<b>Underline</b>	Type Y if you want the field to be underlined.
<b>Truncate</b>	Type Y if you want the field to only partially appear when it is longer than the physical screen. The remainder of the field is truncated and a horizontal scroll is necessary to view it.



## Expanding Definitions

### Literal Text

When defining a screen table, you can display the attributes for the literal text in your screen table. To expand a particular piece of literal text, complete the following tasks from the Screen Table Painter:

1. Place your cursor on the literal text.
2. Press PF15.

A view-only screen showing all the attributes for the literal text appears. The display attributes are the same for all the text literals in the screen table.

```

COMMAND ==>
-----
Literal: Y          Definition for FIELD:
Row : 4          Col: 6
Type:          Syntax: C      Length: 8          Decimal Places: 0          Justify: L

Display Mask:
Globalfield Name :
Value : EMPLOYEE

Visual Attributes
  Foreground Colour :          Highlight : Y
  Background Colour :          Show      :
  Blink             : N        Truncate  :
  Reverse           : N
  Underline         : N
-----
ROW COL      FIELD NAME      Type Syn Len Dec Just Fill Prot Show Rqd Hi Skip Null
---
1  16  MONTH                D   B   8   0   L   .   Y   Y   N   N   Y   Y
1  49  DEPTNO                I   P   2   0   L           Y   Y   N   N   Y   Y
2  49  MANAGER_NUM          I   P   5   0   L           Y   Y   N   N   Y   Y

PFKEYS: 3=SAVE 12=CANCEL

```

### Field Definitions

When defining a screen table, you can expand the definition of a field. This feature invokes a screen that displays more information on a particular field, enabling you to change the field's attributes. It is the same screen you get when you expand on a literal except that you are able to edit most of the attributes for the fields.

The only attributes you cannot edit are the FIELD NAME, Literal, and Globalfield Name. Refer to [Creating New Fields on page 41](#) and [Controlling Presentation on page 44](#) for a description of each of the attributes.

## Expanding a Field

The field you expand is highlighted in the list of fields in the bottom portion of the screen. To expand on a field, complete the following tasks from the Screen Table Painter:

1. Place the cursor on the field.  

You can position the cursor on the field in either the image area or the field definition area.
2. Press PF15.

## Expand Screen Illustrated

The following example illustrates the Expand screen for the field **MONTH**:

COMMAND ==>										Scroll:				
-----														
Literal: N		Definition for FIELD: MONTH												
Row : 1	Col: 16													
Type: D	Syntax: B	Length: 6	Decimal Places: 0				Justify: L							
Display Mask: MMM YY														
Globalfield Name :														
Value :														
Visual Attributes														
Foreground Colour :										Highlight : N				
Background Colour :										Show : Y				
Blink :										Truncate : Y				
Reverse :														
Underline :														
-----														
ROW	COL	FIELD NAME	Type	Syn	Len	Dec	Just	Fill	Prot	Show	Rqd	Hi	Skip	Null
-----														
1	16	MONTH	D	B	8	0	L	.	Y	Y	N	N	Y	Y
1	49	DEPTNO	I	P	2	0	L		Y	Y	N	N	Y	Y
2	49	MANAGER_NUM	I	P	5	0	L		Y	Y	N	N	Y	Y
PFKEYS: 3=SAVE 12=CANCEL														
-----														

## Editing Screen Tables and Screen Fields

---

This section describes the editing functions you can use when editing screen tables and screen fields from the Screen Table Painter. The following editing functions are available:

- Adding a line or screen field
- Copying lines or literals
- Cutting fields or literals
- Pasting fields, lines, or literals
- Deleting lines or fields
- Deleting screen tables

### Adding a Line or Field

To add a blank line or screen field press PF4. Depending on the location of your cursor, the following occurs:

- From the image area, if there is sufficient room, a blank line is added at the line where the cursor is positioned.
- From the field definition area, if there is sufficient room, a new field is added at the line where your cursor is positioned.

### Copying a Line or Literal

PF20 copies the line or literal where the cursor is positioned and holds it until you paste it in a new location.

### Cutting a Field or Literal

PF5 cuts the field or literal where the cursor is positioned and holds it until you paste it in a new location. You can cut a field in either area and paste it in the image area.

### Pasting a Field, Line, or Literal

You can paste a field, line, or literal that was cut by doing the following:

1. Cut or copy the field, line, or literal that you want to paste.

Refer to [Copying a Line or Literal](#) and [Cutting a Field or Literal](#) above for information on cutting or copying fields, lines, and literals.

2. Position the cursor where you want the field, line, or literals to be pasted.
3. Press PF17.
  - If used in the image area, the field(s), line, or literal(s) is pasted where the cursor appears.
  - If used in the field definition area, the field(s) is pasted after the field where the cursor is positioned. If pasting a line, the contents are placed after the line where the cursor is positioned. If there is insufficient room, the paste is not performed and the held content is saved for subsequent pasting.

## Deleting a Line or Field

You can delete information using two different methods:

- PF16
- PF18

You can delete a field using PF18 or PF16 only. If you type over a field in the image area, you are not able to save the screen table. It is restored to the state it was in before being edited when you cancel from the Screen Table Painter.

**Using PF16** Use PF16 to delete the screen fields and/or literal text, depending on the location of your cursor:

- From the image area, PF16 deletes the literal text and screen fields on the line where the cursor is positioned.
- From the field definition area, PF16 deletes only the field where the cursor is positioned. It does not affect literal text or screen fields.

**Using PF18** Use PF18 to delete the field where the cursor is positioned, in either area, or a literal from the image area.

## Deleting a Screen Table

To delete a screen table, you can use one of the following two methods:

- **D** line command from the Screen Definer screen. This deletes the screen table only from the screen definition.
- PF23 and PF22 from the Screen Table Painter. This deletes the screen table both from the screen definition and from TIBCO Object Service Broker.

**Using the D Line Command**

The **D** line command does not delete the screen table definition from TIBCO Object Service Broker. It just removes the screen table from the screen definition. The screen table is still available for use by other screens.

To delete a screen table(s) using the **D** line command, complete the following tasks:

1. Ensure that you are in the Screen Definer screen.  
The screen tables are listed in the Screen Tables section of the screen.
2. Type **D** in the line command field next to the screen table(s) you want to delete.
3. Press Enter.

The screen table(s) is deleted from the screen definition.

**Using PF23 and PF22**

To delete a screen table using PF23 and PF22, complete the following tasks:

1. Ensure that you are in the Screen Table Painter.
2. Press PF23.

This checks the latest version of the TIBCO Object Service Broker Cross Reference Index to determine if any other screens contain the screen table you are about to delete. Pressing PF23 causes one of the following to happen:

— If no other screens use the screen table, a message similar to the following appears:

```
No other Screens referenced "Expense_Data" (as of Mar 24, 2001).
```

— If other screens do use the screen table, a separate screen appears displaying a list of the screens that use the screen table you are about to delete.

PF23 is only active from the Screen Table Painter.

3. Press PF22.

PF22 deletes the screen table from the screen definition, and deletes the screen table definition from TIBCO Object Service Broker. You are prompted to confirm the deletion.



## Chapter 4

# Adding Additional Features to a Screen

This chapter describes how to add additional features to a screen such as Help, Color, or Virtual Fields.

## Topics

---

- [Adding Help Information, page 54](#)
- [Color Implementation, page 60](#)
- [Adding Virtual Fields to a Screen Table, page 63](#)

## Adding Help Information

---

### Overview of Help

TIBCO Object Service Broker provides users with three levels of help: field, screen, and general. In the Screen Definer, via a PF key, you can add the following help to the screen while you are defining it:

- Screen
- Field

### Adding Screen Help

Screen help provides information on the tool. It is accessed when the cursor is not placed on a field or by pressing PF1 again after accessing field help.

To add screen help, complete the following tasks:

1. From the Screen Definer screen, press PF19.

A screen for text editing appears.

2. Add or modify screen help.

You can use the TIBCO Object Service Broker [SCRIPT](#) tool and the [TEXTSETUP](#) HELP to format the help.

3. Press PF5 to script the help text.

This displays the formatted text users see when they press PF1 from the displayed screen.

Toggle back and forth to edit and view.

### Examples of Screen Help

The following two examples illustrate the unformatted Screen Help screen and then the corresponding formatted text similar to what users see when they use PF1.



## Unformatted Screen Help

The following example illustrates unformatted screen help:

---

```
Enter HELP text for screen EMPLOYEE_EXPENSE
-----
_ .setup help
_ .p.This screen is used to enter monthly employee expense data.
_ You can use the following PF keys:
_ .table FCNKEYS(employee_expense), pf_key, name, description
```

```
PFKEYS: 12=QUIT 13=PRINT 3=SAVE 5=SCRIPT
```

---

Formatted Screen Help

The following example illustrates formatted screen help similar to what users see when they use PF1.

Formatted Output		Scroll: P
This screen is used to enter monthly employee expense data. You can use the following PF keys:		
PF_KEY	NAME	DESCRIPTION
PF12	EXIT	EXIT
PF13	PRINT	PRINT
PF3	SAVE	SAVE
PF9	TOTAL	TOTAL
PFKEYS: 12=QUIT 13=PRINT 3=SAVE 5=EDIT		

Adding Field Help

Field help is information about the field where the cursor is currently located.

To add field help, complete the following tasks from the Screen Table Painter:

- 1. Position the cursor on the field to which you want to add help.
- 2. Press PF19.

A screen for text editing appears.

- 3. Add or modify field help.

You can use the TIBCO Object Service Broker [SCRIPT](#) tool and the [TEXTSETUP](#) HELP to format the help.

- 4. Press PF5 to script the help text.

This displays the formatted text as it appears to users when they press PF1 from the displayed screen.

Toggle back and forth to edit and view.

## Examples of Field Help

The following are examples of an unformatted Field Help screen and then the corresponding formatted help similar to what users see when they press PF1.

### Unformatted Field Help

The following example illustrates unformatted field help:

---

```
Enter HELP text for field  MONTH
-----
- .setup help
- .p.Enter the date in the format of MMM YYYY.
```

PFKEYS: 12=QUIT 13=PRINT 3=SAVE 5=SCRIPT

---

Formatted Field Help

The following example illustrates formatted field help similar to what users see when they press PF1:

Formatted Output	Scroll: P
Enter the date in the format of MMM YYYY.	
PFKEYS: 12=QUIT 13=PRINT 3=SAVE 5=EDIT	



Field help appears on the same screen as the prompt options. Every time PF1 is pressed a list of options for that field appears. You can use this feature to provide instructions or explanations of options for prompt fields.

## Example

The following is an example of field-level help (containing user prompting) for the **TYP** field of the Table Definer:

---

```
----- FIELD Level Help -----
COMMAND ==>                                Scroll:

TYP (Mandatory):  Enter the semantic data type of the field you are creating.
The semantic type determines how the data can be used.  The use of this field
ensures that the named field will contain data only of the kind for which it
was intended.  The default is null.

      TYPE_CHAR TYPE_NAME
-   C           Count
-   D           Date
-   I           Identifier
-   L           Logical
-   Q           Quantity
-   S           String

S=Select
PFKEYS: 3=RETURN VALUE 12=CANCEL
```

---

See Also     *TIBCO Object Service Broker Shareable Tools* for more information on the tools.

# Color Implementation

## Adding Color

You can specify background and foreground colors for screen fields and literal text when painting a screen table. You can type the name of the color directly into the field or a prompt screen appears with a list of valid colors.

Two TIBCO Object Service Broker tables and a shareable tool are used to implement the use of color in the Screen Definer. The tables @COLOURS and @SCREENCOLOURS are explained below.

## Purpose of the @COLOURS Table

The @COLOURS table contains a listing of all the colors supported by TIBCO Object Service Broker, independent of the display device. Each color has a corresponding code that uniquely identifies it and a corresponding default code that maps to one of the eight base colors supported on most display devices.

### Example

The following example illustrates of the @COLOURS table:

```

BROWSING TABLE      :  @COLOURS
COMMAND ==>

                                                                    SCROLL: P
      NAME                CODE  DEFAULTCODE                COLORNAME
-----
_  ALICEBLUE                17          1      Alice Blue
_  ANTIQUEWHITE             18          7      Antique White
_  AQUAMARINE               19          5      Aqua Marine
_  AZURE                     20          8      Azure
_  BEIGE                     21          8      Beige
_  BISQUE                    22          8      Bisque
_  BLACK                     8           8      Black
_  BLANCHEDALMOND           23          8      Blanched Almond
_  BLUE                      1           1      Blue
_  BLUEVIOLET                28          1      Blue Violet
_  BLUE1                     24          1      Blue 1
_  BLUE2                     25          1      Blue 2
_  BLUE3                     26          1      Blue 3
_  BLUE4                     27          1      Blue 4
_  BROWN                     13          2      Brown
_  BROWN1                    29          2      Brown 1
_  BROWN2                    30          2      Brown 2

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

```

## Fields in the @COLOURS Table

The fields in the @COLOURS table contain the following information:

NAME	A listing of the colors supported on any device.
CODE	The global color code assigned to each color.
DEFAULTCODE	The default code assigned to each color. If a display device does not support a color, this code maps the unsupported color to one of the eight base colors supported on all devices.
COLORNAME	Lists the color names in mixed case with spaces.

## Purpose of the @SCREENCOLOURS Table

The @SCREENCOLOURS table, parameterized by display device, lists the colors supported for a specific device (that is, 3270). The table has a minimum listing of eight colors. It also contains a field indicating which internal device code is substituted for a color that the device does not support.

### Example

The following example illustrates the @SCREENCOLOURS table instances for a 3270 display device:

```

BROWSING TABLE      :  @SCREENCOLOURS(3270)
COMMAND ==>

                                                                    SCROLL: P

GLOBALCODE  DEVICECODE
-----
-           1           1
-           2           2
-           3           3
-           4           4
-           5           5
-           6           6
-           7           7
-           8           1
-

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

```

## Fields in the @SCREENCOLOURS Table

The fields in the @SCREENCOLOURS table contain the following information:

GLOBALCODE	Color code listing for the global colors supported by the device. Always contains a minimum of eight base colors.
DEVICECODE	An internal device code used to map each color that is not supported on that device to a color that is supported. In the example, the color with a color code of 8 (BLACK), which is not supported on 3270 terminals, is mapped to the internal device code 1 (BLUE), which is supported.

See Also     The [\\$SETCOLOUR](#) tool in *TIBCO Object Service Broker Shareable Tools*.



## Adding Virtual Fields to a Screen Table

---

### Using Virtual Fields

There are two virtual fields contained in each screen table, \$MODIFIED and \$OCCNUM. There are two ways to use these fields:

- Associate these fields with each screen table for use with selection in rules. When used in selection, you cannot order on either of these fields.
- Use these fields in a screen table definition. If used in a screen table definition, they act as described in the definition and not as described below.

### \$MODIFIED Field

\$MODIFIED is used to determine which occurrences in a screen table have been modified. \$MODIFIED has semantic data type L (logical), syntax C (character), and length 1. It behaves as follows:

- When a screen table first appears, \$MODIFIED is set to N (no) for each occurrence.
- When you modify an occurrence by typing into a field, \$MODIFIED is set to Y (yes).
- When the screen table reappears, \$MODIFIED for each occurrence is set to N (no).

### Example

The following example shows how \$MODIFIED can be used in a rule. It returns all occurrences modified by the user.

---

```
DISPLAY EMPLOYEE_EXPENSE ;
FORALL EXPENSE_DATA ( 'EMPLOYEE_EXPENSE' ) WHERE $MODIFIED = 'Y' :
END ;
```

---

## \$OCCNUM Field

\$OCCNUM returns the current position of an occurrence in a screen table (1 is given as the first position). \$OCCNUM has syntax B (binary) and length 4. It positions the screen table buffer on the occurrence where the cursor is positioned.

### Example

The following example shows how \$OCCNUM can be used. It positions the screen table buffer on the occurrence where the cursor is currently placed.

---

```
DISPLAY EMPLOYEE_EXPENSE ;  
GET EXPENSE_DATA ( 'EMPLOYEE_EXPENSE' )  
WHERE $OCCNUM = CURSOROCC# ( 'EMPLOYEE_EXPENSE' ) ;
```

---

## Chapter 5

# Choosing a Menu Style

This chapter defines Menus and their different styles in the context of TIBCO Object Service Broker.

### Topics

---

- [Menus, page 66](#)
- [Choosing a Menu Style, page 67](#)

## Menus

---

### What is a Menu?

A menu is a list of options, created through the [DISPLAY\\_MENU](#) tool, that are presented on a screen. A menu can be part of an application, it can be a login screen for users, or it can be an individualized login screen. You can also use menus to package the components of an application together.

### Menu Styles

There are four menu styles available when you use the [DISPLAY\\_MENU](#) tool from the EX execute rule option on the workbench:

- Menu Manager menu
- Screen Manager menu
- Selection Manager menu
- Standard Session Manager menu

## Choosing a Menu Style

Before using the [DISPLAY\\_MENU](#) tool to create your menu, you should choose a menu style. To do this, you must decide what your requirements are for the menu you want to define.

### How to Choose a Menu Style

The following table lists possible requirements you could have when defining a menu and the best menu style to use for that particular requirement.

If you...	Menu to Use	Refer to...
Require two or more columns	Selection Manager	<a href="#">Chapter 8, Creating a Selection Manager Menu, on page 87</a>
Want to incorporate shareable design elements	Screen Manager	<a href="#">Chapter 7, Creating a Screen Manager Menu, on page 77</a>
Require a command line	Standard Session Manager	<a href="#">Chapter 9, Creating a Standard Session Manager Menu, on page 97</a>
Have a predefined screen	Screen Manager	<a href="#">Chapter 7, Creating a Screen Manager Menu, on page 77</a>
Require only one column	Menu Manager	<a href="#">Chapter 6, Creating a Menu Manager Menu, on page 69</a>
Require the command history area	Standard Session Manager	<a href="#">Chapter 9, Creating a Standard Session Manager Menu, on page 97</a>
Require the appointment calendar	Standard Session Manager	<a href="#">Chapter 9, Creating a Standard Session Manager Menu, on page 97</a>
Want to use a TIBCO Object Service Broker screen	Selection Manager	<a href="#">Chapter 8, Creating a Selection Manager Menu, on page 87</a>

If you...	Menu to Use	Refer to...
Require browse and test flags	Standard Session Manager	<a href="#">Chapter 9, Creating a Standard Session Manager Menu, on page 97</a>
Require the library name on the menu	Standard Session Manager	<a href="#">Chapter 9, Creating a Standard Session Manager Menu, on page 97</a>
Require date and time fields	Standard Session Manager	<a href="#">Chapter 9, Creating a Standard Session Manager Menu, on page 97</a>

## Chapter 6

# Creating a Menu Manager Menu

This chapter describes how to create a Menu Manager Menu.

### Topics

---

- [Menu Manager Menu, page 70](#)
- [Defining the Physical Layout, page 71](#)
- [Adding Functionality to a Menu, page 74](#)

## Menu Manager Menu

---

The Menu Manager menu style enables you to create a simple one column listing of menu item fields, each of which a user can select. A predefined screen is not required.

This menu style can be used as a login menu, be nested within another menu, or have other menus nested within it.

### Menu Manager Menu Illustrated

The following example illustrates a Menu Manager menu:

---

Expense reporting for USR40 as of 03-14-2000

Create a report \_  
Query a report \_\_\_\_\_

Press ENTER beside the option desired

PFKEYS: 2=LOGS 3=EXIT 12=EXIT

---

### Procedure

To create this menu, complete the following tasks:

1. [Defining the Physical Layout, page 71](#)
2. [Adding Functionality to a Menu, page 74](#)



# Defining the Physical Layout

Complete the following tasks to define the physical layout of a Menu Manager menu:

- 1. [Access the Menu Definer tool, page 71](#)
- 2. [Specify titles, page 72](#)
- 3. [Add menu items, page 73](#)

These tasks are described in detail in the sections below.

### Example

The following example illustrates the Menu Manager definition screen for the Expense reporting for menu (refer to [Menu Manager Menu Illustrated on page 70](#)):

Command ==>				Define	Menu: MENU_EMPLOYEE	Unit: USR40
L#	T/B	Just	Hi	Titles for Screen		
1	T	C	Y	Expense reporting for .userid as of .date 'mm-dd-yyyy'		
1	B	C	Y	Press ENTER beside the option desired		
-----						
Entries (L/R): R				Menu Items		
Time & Ok: Y						
Title				Rule	Action	Search Browse
					New/Call/Menu	S/I/L Y/N
Create a report					M	-
Query a report					M	-
PFKEYS: 3=SAVE 12=CANCEL 4=+LINE 16=-LINE 5=+TITLE 17=-TITLE 6=TEST 2=DOC						

### Task A Access the Menu Definer tool

To access the Menu Definer to create a Menu Manager menu, use the following tasks:

- 1. From the EX execute rule option on the workbench, type `DEFINE_MENU(menuname)`.

If you do not include anything in the parentheses, a list of existing menus appears from which you can choose one to edit. Type **S** beside the menu you want to edit and press Enter. The appropriate definition screen for the chosen menu appears.

When using `DEFINE_MENU`, TIBCO Object Service Broker must run in update mode (Browse = N).

- 2. Press Enter.

The appropriate menu definition screen appears.

If the menu is new, an option list of menu styles appears on the SELECT TYPE OF MENU TO BE DEFINED screen. Type **S** beside the Menu Manager menu option you want and press PF3. A blank definition screen appears.

**Task B Specify titles**

You use the ‘Titles for Screen’ portion of the screen to specify the titles you want to appear on the menu. You can stipulate up to three title lines for both the top and the bottom of a menu. You can also specify up to three entries for each title line: one for the left side, one for the center, and one for the right side of the menu.

The following fields are used to specify titles for the menu. For valid values, press PF1.

<b>L#</b>	Specify the position of the title line on the screen by indicating at what line the title occurs. The position of the title line is in relation to the other lines.
<b>T/B</b>	Indicate whether the title line appears at the top (T) or bottom (B) of the menu.
<b>Just</b>	Specify the position of the text on the title line. You can specify if the text is left, right, or center justified.
<b>Hi</b>	Specify if the text on the title line is highlighted (Y).
<b>Titles for Screen</b>	Type any literal text in this field. This literal text appears on the title line of the menu. The present date and the user’s user ID can also be included here. For valid display mask components that you can use with the <code>.date</code> command, refer to <a href="#">Display Masks on page 122</a> .

**Task C   Add menu items**

You use the Menu Items portion of the screen to specify the options that appear on the menu and how they function. The following fields are used to add menu items. For valid values, press PF1.

<b>Entries (L/R)</b>	Specify on which side of the menu item you want the user entry field to appear. Currently, only display to the right of the menu item is supported.
<b>Time &amp; Ok</b>	Indicate whether the transaction displays the current time and the message OK at the bottom of the menu. You do not have to display a message.
<b>Title</b>	Type a description of the menu item. This description appears as an item on the menu.

## Adding Functionality to a Menu

After defining the physical layout of the menu, you must determine the functionality of each menu item. There are three options for a menu item, it can:

- Start a new transaction
- Call a rule in the same transaction
- Display another menu

Determine which function for a particular menu item you require and refer to the appropriate section below.

### Starting a New Transaction

The following fields are used to define a menu item that starts a new transaction. For valid values, press PF1.

<b>Rule</b>	Type the name of the rule to be invoked when the user selects this item from the menu.
<b>Action New/Call/Menu</b>	Type N for NEW. This indicates that the menu item starts a new transaction when invoked.
<b>Search S/I/L</b>	Specify the level of library to start searching for the designated rule.
<b>Browse Y/N</b>	Set to Y to execute the rule in browse mode.
<b>Parm Y/N</b>	Specify whether (Y) the user can enter arguments for the rule.
<b>Prompt Obj/Parm/None</b>	If the <b>Parm Y/N</b> field is set to Y, specify the way in which the user is prompted. If you specify N and the user does not enter anything, the rule is passed a null for each of its arguments.
<b>Entry Length</b>	Specify a value for the maximum length of information the user can enter in the prompting field. The length of the prompting field depends on the argument(s) for the specified rule.

<b>Fill</b>	Type a character to be used as a fill character. The character appears repeatedly to the maximum length of the <b>Entry Length</b> field.
<b>Obj list or Menu</b>	<p>If the <b>Prompt</b> field is set to O, type the name of the object list. The object list appears in the <b>Obj list or Menu</b> field.</p> <p>The name you enter must be the same as the table name used when you defined the object list, using the <a href="#">DEFINE_OBJLIST</a> tool. You do not have to predefine the object list; it must be defined by the time you want to use <a href="#">DEFINE_MENU</a> or use the menu at login time. Refer to <i>TIBCO Object Service Broker Shareable Tools</i> for more information on the <a href="#">DEFINE_OBJLIST</a> tool and <a href="#">Appendix A, Displaying an Application, on page 115</a> for more information on the <a href="#">DISPLAY_MENU</a> tool.</p>
<b>Usage</b>	Type a comment line. This line explains the use of the menu item.
<b>Purpose</b>	Type the purpose of the menu item.

## Calling a Rule

The following fields are used to define a menu item that calls a TIBCO Object Service Broker rule. For valid values, press PF1. Some fields are not supported for a selection that calls a rule, for example, you cannot set the browse mode or search path.

<b>Rule</b>	Type a rule name. The rule is invoked when the user selects the item from the menu.
<b>Action New/Call/Menu</b>	Type C for CALL. This indicates that the menu item calls a TIBCO Object Service Broker rule. If you call a rule, you cannot use this menu as a login menu.
<b>Parm Y/N</b>	Type N. The rule called cannot have arguments.
<b>Prompt Obj/Parm/None</b>	Type N. The rule cannot have arguments.
<b>Usage</b>	Type a comment line. This line explains the use of the menu item.

<b>Purpose</b>	Type the purpose of the menu item.
----------------	------------------------------------

Calling Another Menu

The following fields are used to define a menu item that calls another menu. For valid values, press PF1. Some fields are not supported for a selection that calls another menu, for example, you cannot specify a search path.

<b>Action New/Call/Menu</b>	Type M for MENU. This indicates that the menu item calls another menu.
<b>Obj list or Menu</b>	Specify the name of the menu to appear. This menu must be defined using <a href="#">DEFINE_MENU</a> . You do not have to predefine the referenced menu; it must be defined before the menu referencing it can be used. Refer to <a href="#">Appendix A, Displaying an Application, on page 115</a> for more information on the <a href="#">DEFINE_MENU</a> tool.
<b>Usage</b>	Type a comment line. This line explains the use of the menu item.
<b>Purpose</b>	Type the purpose of the menu item.

## Chapter 7      **Creating a Screen Manager Menu**

This chapter describes how to create a Screen Manager Menu.

### Topics

---

- [Screen Manager Menu, page 78](#)
- [Defining the Physical Layout, page 79](#)
- [Adding Functionality to a Menu, page 83](#)

## Screen Manager Menu

---

The Screen Manager menu style enables you to create a simple one column listing of menu item fields, each of which a user can select. This menu style can be used as a login menu, be nested within another menu, or have other menus nested within it.

### Predefined Screen Requirements

You must have a screen predefined for the menu. Therefore, you can incorporate shareable design elements into the menu.

### Screen Manager Menu Illustrated

The following example illustrates a Screen Manager menu:

---

Software Development Corporation

- \_ Create an expense report
- \_ Create a status report

Press ENTER beside the option desired

PFKEYS: 2=LOGS 3=EXIT 12=EXIT

---

### Procedure

To create this menu, complete the following tasks:

- [Defining the Physical Layout, page 79](#)
- [Adding Functionality to a Menu, page 83](#)



# Defining the Physical Layout

The following tasks are required to define the physical layout of a Screen Manager menu:

- 1. [Predefine a screen, page 80](#)
- 2. [Access the Menu Definer tool, page 81](#)
- 3. [Define the physical layout of the menu, page 81](#)
- 4. [Add menu items, page 82](#)

These tasks are described in detail in the sections below.

## Example

The following example illustrates the Screen Manager menu definition screen for the Software Development Corporation menu (refer to [Screen Manager Menu Illustrated on page 78](#)):

Define

Menu: SCR\_EMPLOYEE

Unit: USR40

Command ==>

Screen to be used: WEEKLY\_REP

Scrolling table containing fields "TITLE" and "VALUE": FIELD

Optional table with field named "DATE": TITLE\_1

-----

Menu Items

Time & Ok: Y

Title

-----

Create an expense report

Create a status report

Rule

-----

WEEKLY\_STA\_R

Action

New/Call/Menu

-----

M

C

Search

S/I/L

-----

-

L

Browse

Y/N

-----

-

N

PFKEYS: 3=SAVE 12=CANCEL 4=+LINE 16=-LINE 22=DEL 6=TEST 9=EDHELP 2=DOC

Task A Predefine a screen

Before creating the Screen Manager menu, you must define a screen to be used with it. The predefined screen can have any layout and contain any titles as long as it contains the following:

- A scrollable screen table, with Max Occ set to asterisk (\*) and the following fields:

Field	Description	Defined as ...
Title	Contains the title for each menu item.	Syntax V Maximum Length 25 Visible Y Protected
Value	Contains the user entries.	Syntax V Long enough for any expected user entry Unprotected

- The screen table PFKEY\_SPECS to contain the PF key specification. This screen table must have the fields **ROW** and **COL** defined to -2 and 1 respectively.
- The screen table SESSMGR\_MSG to contain the message line. This screen table must have the fields **ROW** and **COL** defined to -1 and 1 respectively.
- An optional screen table with a field called **DATE** if you want to display the current date. This field must be defined as semantic type D, protected, and visible (**Show** must be set to Y).

To use a date format other than the default format YYYY-MM-DD, you must define a display mask for it in the **Display Mask** field of the Screen Definer. Refer to [Display Masks on page 122](#) for more information on display masks.

Refer to [Chapter 2, Building a Screen, on page 7](#) for more information on defining a screen and [Chapter 3, Defining Screen Tables, on page 25](#) for more information on defining a screen table.



Do not use PF18 from the Screen Definer to define your FCN keys or PF keys. The menu that you create with the Menu Definer uses the predefined PF keys 2, 3, and 12; you cannot define function keys for a menu.

**Task B Access the Menu Definer tool**

To access the Menu Definer to create a Screen Manager menu, complete the following tasks:

- 1. From the EX execute rule option on the workbench, type `DEFINE_MENU(menuname)`.

If you do not include anything in the parentheses, a list of existing menus appears from which you can choose one to edit. Type **S** beside the menu you want to edit and press Enter. The appropriate definition screen for the chosen menu appears.

When using `DEFINE_MENU`, TIBCO Object Service Broker must run in update mode (Browse = N).

- 2. Press Enter.

The appropriate menu definition screen appears.

If the menu is new, an option list of menu styles appears on the SELECT TYPE OF MENU TO BE DEFINED screen. Type **S** beside the Screen Manager menu option and press PF3. A blank definition screen appears.

**Task C Define the physical layout of the menu**

The following fields are used to define the physical layout of the Screen Manager menu. For valid values, press PF1:

<b>Screen to be used</b>	Type the name of a predefined screen. This screen determines the physical layout of the menu.
<b>Scrolling table containing fields TITLE and VALUE</b>	Type the name of the scrollable screen table containing the fields <b>TITLE</b> and <b>VALUE</b> .
<b>Optional table with field named DATE</b>	<p>To display the current date, type the name of a screen table that has a field named <b>DATE</b>. The <b>DATE</b> field must be defined as semantic type D, protected, and visible (<b>Show</b> must be set to Y).</p> <p>To use a date format other than the default of YYYY-MM-DD, you must define a display mask for it in the <b>Display Mask</b> field of the Screen Definer. For more information on display masks, refer to <a href="#">Display Masks on page 122</a>.</p>

**Task D Add menu items**

The following fields are used to add menu items to the menu. For valid values, press PF1.

<b>Time &amp; Ok</b>	Indicate whether the transaction displays the current time and the message OK at the bottom of the menu. You do not have to display a message.
<b>Title</b>	Type a description of the menu item. This description appears as an item on the menu.

## Adding Functionality to a Menu

After defining the physical layout of the menu, you must determine the functionality of each menu item. There are three options for a menu item, it can:

- Start a new transaction
- Call a rule in the same transaction
- Display another menu

Determine which function for a particular menu item you require and refer to the appropriate section below.

### Starting a New Transaction

The following fields are used to define a menu item that starts a new transaction. For valid values, press PF1.

<b>Rule</b>	Type the name of the rule to be invoked when the user selects this item from the menu.
<b>Action New/Call/Menu</b>	Type N for NEW. This indicates that the menu item starts a new transaction when invoked.
<b>Search S/I/L</b>	Specify the level of library to start searching for the designated rule.
<b>Browse Y/N</b>	Set to Y to execute the rule in browse mode.
<b>Parm Y/N</b>	Specify whether (Y) the user can enter arguments for the rule.
<b>Prompt Obj/Parm/None</b>	If the <b>Parm Y/N</b> field is set to Y, specify the way in which the user is prompted. If you specify N and the user does not enter anything, the rule is passed a null for each of its arguments.
<b>Entry Length</b>	Specify a value for the maximum length of information the user can enter in the prompting field. The length of the prompting field depends on the argument(s) for the specified rule.

<b>Fill</b>	Type a character to be used as a fill character. The character appears repeatedly to the maximum length of the <b>Entry Length</b> field.
<b>Obj list or Menu</b>	<p>If the <b>Prompt</b> field is set to O, type the name of the object list. The object list appears in the <b>Obj list or Menu</b> field.</p> <p>The name you enter must be the same as the table name used when you defined the object list, using the <a href="#">DEFINE_OBJLIST</a> tool. You do not have to predefine the object list; it must be defined by the time you want to use <a href="#">DEFINE_MENU</a> or use the menu at login time. Refer to <i>TIBCO Object Service Broker Shareable Tools</i> for more information on the <a href="#">DEFINE_OBJLIST</a> tool and <a href="#">Appendix A, Displaying an Application, on page 115</a> for more information on the <a href="#">DEFINE_MENU</a> tool.</p>
<b>Usage</b>	Type a comment line. This line explains the use of the menu item and can be included in the help for your menu. Refer to <a href="#">Adding Help Information on page 112</a> for information on adding help to menus.
<b>Purpose</b>	Type the purpose of the menu item. You can include this field in the help for your menu. Refer to <a href="#">Adding Help Information on page 112</a> for information on adding help to menus.

Calling a Rule

The following fields are used to define a menu item that calls a TIBCO Object Service Broker rule. For valid values, press PF1. Some fields are not supported for a selection that calls a rule, for example, you cannot set the browse mode or search path.

<b>Rule</b>	Type a rule name. The rule is invoked when the user selects the item from the menu.
<b>Action New/Call/Menu</b>	Type C for CALL. This indicates that the menu item calls a TIBCO Object Service Broker rule. If you call a rule, you cannot use this menu as a login menu.
<b>Prompt Obj/Parm/None</b>	Type N. The rule cannot have arguments.

<b>Parm Y/N</b>	Type N. The rule called cannot have arguments.
<b>Usage</b>	Type a comment line. This line explains the use of the menu item and can be included in the help for your menu. Refer to <a href="#">Adding Help Information on page 112</a> for information on adding help to menus.
<b>Purpose</b>	Type the purpose of the menu item. You can include this field in the help for your menu. Refer to <a href="#">Adding Help Information on page 112</a> for information on adding help to menus.

## Calling Another Menu

The following fields are used to define a menu item that calls another menu. For valid values, press PF1. Some fields are not supported for a selection that calls another menu, for example, you cannot specify a library level.

<b>Action New/Call/Menu</b>	Type M for MENU. This indicates that the menu item calls another menu.
<b>Obj list or Menu</b>	Specify the name of the menu to appear. This menu must be defined using <code>DEFINE_MENU</code> . You do not have to predefine the referenced menu; it must be defined before the menu referencing it can be used. Refer to <a href="#">Appendix A, Displaying an Application, on page 115</a> for more information on the <code>DEFINE_MENU</code> tool.
<b>Usage</b>	Type a comment line. This line explains the use of the menu item and can be included in the help for your menu. Refer to <a href="#">Adding Help Information on page 112</a> for information on adding help to menus.
<b>Purpose</b>	Type the purpose of the menu item. You can include this field in the help for your menu. Refer to <a href="#">Adding Help Information on page 112</a> for information on adding help to menus.





## Chapter 8      **Creating a Selection Manager Menu**

This chapter describes how to create a Selection Manager Menu.

### Topics

---

- [Selection Manager Menu, page 88](#)
- [Defining the Physical Layout, page 90](#)
- [Adding Functionality to a Menu, page 94](#)

## Selection Manager Menu

---

The Selection Manager menu style enables you to create a more complex listing of menu items, each of which the user can select, than the Screen Manager menu style.

The Selection Manager menu is typically used when the listing of options requires more than one column of display. It can be used as a login menu, be nested within another menu, or have other menus nested within it.

### Predefined Screen Requirement

The Selection Manager menu has no screen attributes that are predefined; however, you must define a TIBCO Object Service Broker screen using the Screen Definer before you can create this menu. The defined screen is then used in conjunction with this menu style.

### Selection Manager Menu Illustrated

The following example illustrates a Selection Manager menu:

---

2000/04/14	Software Development Corporation	Unit: USR40
_ Weekly Expenses		
_ Weekly Status Report    _ open    _ closed		
Press ENTER beside the option desired		
PFKEYS: 2=LOGS 3=EXIT 12=EXIT		

---

## Procedure

To create this menu, complete the following tasks:

- [Defining the Physical Layout, page 90](#)
- [Adding Functionality to a Menu, page 94](#)

## Defining the Physical Layout

The following tasks are required to define the physical layout of the Selection Manager menu:

- 1. [Predefine a screen, page 91](#)
- 2. [Access the Menu Definer tool, page 92](#)
- 3. [Define the physical layout, page 92](#)
- 4. [Add menu items, page 93](#)

These tasks are described in detail in the sections below.

### Example

The following example illustrates the Selection Manager Definition screen for the Software Development Corporation menu (refer to [Selection Manager Menu Illustrated on page 88](#)):

Define Menu: SEL\_EMPLOYEE

Unit: USR40

Command ==>

Screen to be used: WEEKLY\_REPORT

Table containing menu fields: SELECTION

Optional table with field named "DATE": TITLE

Optional table with field named "USERID": TITLE

-----

Menu Items

Time & Ok: Y

Field Name

-----

EXPENSES

STATUS

Rule

-----

WEEKLY\_REPORT\_RU

WEEKLY\_STATUS\_RU

Action

New/Call/Menu

-----

N

N

Search

S/I/L

-----

L

L

Browse

Y/N

-----

N

N

PFKEYS: 3=SAVE 12=CANCEL 4=+LINE 16=-LINE 22=DEL 6=TEST 9=EDHELP 2=DOC

Task A Predefine a screen

Before creating the Selection Manager menu, you must define a screen to be used with it. The predefined screen can have any layout and contain any titles as long as it contains the following:

- A screen table that has fields defined for each menu item. Each menu item field must have a title and be defined as:

Semantic Data Type	S
Syntax	C
Length	Long enough for any expected user entry
Protected	No
Visible	Yes

- The screen table PFKEY\_SPECS to contain the PF key specifications. This screen table must have the fields **ROW** and **COL** defined to -2 and 1 respectively.
- The screen table SESSMGR\_MSG to contain the message line. This screen table must have the fields **ROW** and **COL** defined to -1 and 1 respectively.
- An optional screen table with a field called **USERID**, if you want the user’s TIBCO Object Service Broker user ID to appear on the menu.
- An optional screen table with a field called **DATE**, if you want to display the current date on the menu. This field must be defined as semantic type D, protected, and visible (**Show** must be set to Y).

To use a date format other than the default format YYYY-MM-DD, you must define a display mask for it in the **Display mask** field of the Screen Definer. Refer to [Display Masks on page 122](#) for more information on display masks.

Refer to [Chapter 2, Building a Screen, on page 7](#) for more information on defining a screen and [Chapter 3, Defining Screen Tables, on page 25](#) for more information on defining a screen table.



Do not use PF18 from the Screen Definer to define your function keys. The menu that you create with the Menu Definer uses the predefined PF keys 2, 3, and 12; you cannot define function keys for a menu.

**Task B Access the Menu Definer tool**

To access the Menu Definer to create a Selection Manager menu, complete the following tasks:

1. From the EX execute rule option on the workbench, type `DEFINE_MENU(menuname)`.

If you do not include anything in the parentheses, a list of existing menus appears from which you can choose. Type **S** beside the menu and press Enter. The appropriate definition screen for the chosen menu appears.

When using `DEFINE_MENU`, TIBCO Object Service Broker must run in update mode (Browse = N).

2. Press Enter.

The appropriate menu definition screen appears.

If the menu is new, an option list of menu styles appears on the SELECT TYPE OF MENU TO BE DEFINED screen. Type **S** beside the Selection Manager menu option and press PF3. A blank definition screen appears.

**Task C Define the physical layout**

The following fields are used to define the physical layout of the menu. For valid values, press PF1.

<b>Screen to be used</b>	Type the name of the predefined screen that determines the physical layout of the menu.
<b>Table containing menu fields</b>	Specify the name of the table containing menu fields.
<b>Optional table with field named "DATE"</b>	To display the current date, type the name of a screen table that has a field named <b>DATE</b> . The field must be defined as semantic data type D, protected, and visible ( <b>Show</b> must be set to Y).
<b>Optional table with field named "USERID"</b>	Type the name of the screen table that contains the <b>USERID</b> field, if required. This displays the TIBCO Object Service Broker user ID of the user.

**Task D Add menu items**

The following fields are used to add menu items to the menu. For valid values, use PF1.

<b>Time &amp; Ok</b>	Indicate whether the transaction displays the current time and the message OK at the bottom of the message. You do not have to display a message.
<b>Field Name</b>	Type the name of a screen field (and a literal text associated with it). This screen appears as an item on the menu and it must exist in the defined screen table.

## Adding Functionality to a Menu

After defining the physical layout of the menu, you have to determine the functionality of each menu item. There are three options for a menu item, it can:

- Start a new transaction
- Call a rule in the same transaction
- Display another menu

Determine which function for a particular menu item you require and refer to the appropriate section below.

### Starting a New Transaction

The following fields are used to define a menu item that starts a new transaction. For valid values, press PF1.

<b>Rule</b>	Type the name of the rule to be invoked when the user selects this item from the menu.
<b>Action New/Call/Menu</b>	Type N for NEW. This indicates that the menu item starts a new transaction when invoked.
<b>Search S/I/L</b>	Specify the level of library to start searching for the designated rule.
<b>Browse Y/N</b>	Set to Y to execute the rule in browse mode.
<b>Parm Y/N</b>	Specify whether (Y) the user can enter arguments for the rule.
<b>Prompt Obj/Parm/None</b>	If the <b>Parm Y/N</b> field is set to Y, specify the way in which the user is prompted. If you specify N and the user does not enter anything, the rule is passed a null for each of its arguments.
<b>Entry Length</b>	Specify a value for the maximum length of information the user can enter in the prompting field. The length of the prompting field depends on the argument(s) for the specified rule.



<b>Fill</b>	Type a character to be used as a fill character. The character appears repeatedly to the maximum length of the <b>Entry Length</b> field.
<b>Obj list or Menu</b>	<p>If the <b>Prompt</b> field is set to O, type the name of the object list. The object list appears in the <b>Obj list or Menu</b> field.</p> <p>The name you enter must be the same as the table name used when you defined the object list, using the <a href="#">DEFINE_OBJLIST</a> tool. You do not have to predefine the object list; it must be defined by the time you want to use <a href="#">DEFINE_MENU</a> or use the menu at login time. Refer to <i>TIBCO Object Service Broker Shareable Tools</i> for more information on the <a href="#">DEFINE_OBJLIST</a> tool and <a href="#">Appendix A, Displaying an Application, on page 115</a> for more information on the <a href="#">DEFINE_MENU</a> tool.</p>
<b>Usage</b>	Type a comment line. This line explains the use of the menu item and can be included in the help for your screen. Refer to <a href="#">Adding Help Information on page 112</a> for information on adding help to menus.
<b>Purpose</b>	Type the purpose of the menu item. You can include this field in the help for your screen. Refer to <a href="#">Adding Help Information on page 112</a> for information on adding help to menus.

## Calling a Rule

The following fields are used to define a menu item that calls a TIBCO Object Service Broker rule. For valid values, press PF1. Some fields are not supported for a selection that calls a rule, for example, you cannot set the browse mode or search path.

<b>Rule</b>	Type a rule name. The rule is invoked when the user selects the item from the menu.
<b>Action New/Call/Menu</b>	Type C for CALL. This indicates that the menu item calls a TIBCO Object Service Broker rule. If you call a rule, you cannot use this menu as a login menu.
<b>Parm Y/N</b>	Type N. The rule called cannot have arguments.

<b>Prompt Obj/Parm/None</b>	Type N. The rule cannot have arguments.
<b>Usage</b>	Type a comment line. This line explains the use of the menu item and can be included in the help for your menu. Refer to <a href="#">Adding Help Information on page 112</a> for information on adding help to menus.
<b>Purpose</b>	Type the purpose of the menu item. You can include this field in the help for your menu. Refer to <a href="#">Adding Help Information on page 112</a> for information on adding help to menus.

Calling Another Menu

The following fields are used to define a menu item that calls another menu. For valid values, press PF1. Some fields are not supported for a selection that calls another menu, for example, you cannot specify a library level.

<b>Action New/Call/Menu</b>	Type M for MENU. This indicates that the menu item calls another menu.
<b>Obj list or Menu</b>	Specify the name of the menu to appear. This menu must be defined using <a href="#">DEFINE_MENU</a> . You do not have to predefine the referenced menu; it must be defined before the menu referencing it can be used. Refer to <a href="#">Appendix A, Displaying an Application, on page 115</a> for more information on the <a href="#">Adding Help Information on page 112</a> tool.
<b>Usage</b>	Type a comment line. This line explains the use of the menu item and can be included in the help for your menu. Refer to <a href="#">Adding Help Information on page 112</a> for information on adding help to menus.
<b>Purpose</b>	Type the purpose of the menu item. You can include this field in the help for your menu. Refer to <a href="#">Adding Help Information on page 112</a> for information on adding help to menus.

## Chapter 9

# Creating a Standard Session Manager Menu

This chapter describes how to create a Standard Session Manager Menu.

## Topics

---

- [Standard Session Manager Menu, page 98](#)
- [Defining the Physical Layout, page 99](#)
- [Adding Functionality to a Menu, page 103](#)

## Standard Session Manager Menu

---

The Standard Session Manager menu style is used to define a user’s login menu when you want to include items such as test and browse flags, the appointment calendar, and the command history area. These screen attributes are predefined and you can selectively include them in the menu. If you do not require these attributes, you can use one of the other menu styles.

The Standard Session Manager menu cannot be nested, however, it can have other menus nested within it.

### Standard Session Manager Menu Illustrated

The following example illustrates a Standard Session Manager menu:

Software DEVELOPMENT CORPORATION									
LIBRARY	USR40	TEST: N	BROWSE: N	1:45	PM	TUESDAY MAR 14 2000			
						SU	MO	TU	WE TH FR SA
BR browse table	==>								1 2 3 4
ED edit table	==>					5	6	7	8 9 10 11
ER edit rule	==>					12	13	14	15 16 17 18
						19	20	21	22 23 24 25
PR print table	==>					26	27	28	29 30 31
COMMAND	==>	___							
PFKEYS: 2=LOGS 3=EXIT 12=EXIT									

### Procedure

To create this menu, complete the following tasks:

- [Defining the Physical Layout, page 99](#)
- [Adding Functionality to a Menu, page 103](#)

## Defining the Physical Layout

The following tasks are required to define the physical layout of the Standard Session Manager menu:

1. Access the Menu Definer tool, page 100
2. Define the session menu area, page 100
3. Define the head area, page 101
4. Define the menu area, page 101
5. Define the command area, page 102
6. Add menu items, page 102

These tasks are described in detail in the sections that follow.

## Example

The following example illustrates the Standard Session Manager Definition screen for the Software Development Corporation menu (refer to [Standard Session Manager Menu Illustrated on page 98](#)):

```

Command ==>
Define Session Menu: STANDARD_EMP
Menu Screen Unit: USR40      Include Calendar/Appointments: Y
Top Line (optional): Software DEVELOPMENT CORPORATION

Head      - Screentable Name: STANDARD_EM_HEAD
Area      Side Title: LIBRARY                                (Length: 16 )
          Library: Y  Test & Browse Flags: Y  Date & Time: Y

Menu      - Screentable Name: STANDARD_EM_BODY
Area      Arrows: Y  Abbreviations: Y  Title Length: 12  Entry Length: 30

Command - Screentable Name: STANDARD_EM_HIST
Area      Command Line: Y  History Length: 50
-----
          Menu Items
Time & Ok: Y
Ab      Title      Rule      Action      Search Browse
          S/I/L      Y/N
--  -----  -
BR browse table      STEBROWSE      N      S      Y
                                   N
                                   N
PFKEYS: 1=HELP 3=SAVE 12=CANCEL 4=+LINE 16=-LINE 6=TEST 9=EDHELP 22=DEL 2=DOC

```

**Task A Access the Menu Definer tool**

To access the Menu Definer to create a Standard Session Manager menu, complete the following tasks:

- 1. From the EX execute rule option on the workbench, type `DEFINE_MENU(menuname)`.

If you do not include anything in the parentheses, a list of existing menus appears from which you can choose one to edit. Type **S** beside the menu you want to edit and press Enter. The appropriate definition screen for the chosen menu appears.

When using `DEFINE_MENU`, TIBCO Object Service Broker must run in update mode (Browse = N).

- 2. Press Enter.

The appropriate menu definition screen appears.

If the menu is new, an option list of menu styles appears on the SELECT TYPE OF MENU TO BE DEFINED screen. Type **S** beside the Standard Session Manager menu and press PF3. A blank definition screen appears.

**Task B Define the session menu area**

The following fields are used to define the session menu. For valid values, press PF1.

<b>Define Session Menu</b>	Modify the name of the menu, if required. You can change the name any time during the menu definition. Refer to <a href="#">Copying Menu Definitions on page 108</a> for more information on copying menus.
<b>Menu Screen Unit</b>	Modify the menu screen unit, if required. This is the user ID of the person creating the menu. You can change the unit any time during the menu definition.
<b>Include Calendar Appointment</b>	Specify whether you want the calendar and appointment book on your menu. If you type Y to include the calendar, the date and time must be enabled in the head area so that the appointment book contains more than the present month.
<b>Top Line (optional)</b>	Type a top line title, if required. This information always appears on the menu centered and in uppercase.

### Task C Define the head area

The following fields are used to define the menu header. For valid values, press PF1.

<b>Screentable Name</b>	Modify the screen table name that controls the menu header, if required. By default, this screen table is given the name of the first eleven characters of the menu suffixed by <b>_HEAD</b> . You can change the name of the screen table as long as a table with the new name does not already exist.
<b>Side Title</b>	Type the title text to appear on the second line of the menu. The maximum length of this title is determined by the other information that appears on the second line (the <b>Library</b> , <b>Test &amp; Browse Flags</b> field, or <b>Date &amp; Time</b> fields). You can change this title at any time during the menu definition.
<b>Library</b>	Specify whether you want the library name to appear on the second line of the menu.
<b>Test &amp; Browse Flags</b>	Determine whether you want the test and browse flags to appear on the menu. These flags enable the user to specify if rules should run in test or browse mode; only specify Y if the item EXECUTERULE is on the menu.
<b>Date &amp; Time</b>	Indicate whether you want the date and time to appear on the menu.

### Task D Define the menu area

The following fields are used to define the menu area of the menu. For valid values, press PF1.

<b>Screentable Name</b>	Modify the screen table name that controls the menu body, if required. By default, this screen table is given the name of the first eleven characters of the menu suffixed by <b>_BODY</b> . You can change this name as long as a table with the new name does not already exist.
<b>Arrows</b>	Specify whether you want menu item arrows to appear beside the menu items.

<b>Abbreviations</b>	Indicate whether you want the menu item abbreviation to appear on the menu. The abbreviation is specified in the <b>Ab</b> field in the Menu Items portion.
----------------------	---

**Task E Define the command area**

The following fields are used to define the command area of the menu. For valid values, press PF1.

<b>Screentable Name</b>	Modify the screen table name that controls the command area, if required. By default, this screen table is given the name of the first eleven characters of the menu suffixed by _HIST. You can change this name as long as a table with the new name does not already exist.
<b>Command Line</b>	Specify whether you want to include the primary command line.
<b>History Length</b>	Type a value to change the default length of the command history area that appears below the primary command line. The default length is 50. To turn the history option off, type zero in the field.

**Task F Add menu items**

The following fields are used to add menu items to the menu. For valid values, press PF1.

<b>Time &amp; Ok</b>	Indicate whether the transaction displays the current time and the message OK at the bottom of the menu. You do not have to display a message.
<b>Ab</b>	Type a two letter abbreviation for the menu item function. This abbreviation is used in the command line as the abbreviation for the application. The abbreviation appears only if you specify Y for <b>Abbreviation</b> in the menu area of the screen.
<b>Title</b>	Type a description of the application. This description appears as the menu item.



## Adding Functionality to a Menu

After defining the physical layout of the menu, you must determine the functionality of each menu item. There are three options for a menu item, it can:

- Start a new transaction
- Call a rule in the same transaction
- Display another menu

Determine which function for a particular menu item you require and refer to the appropriate section below.

### Starting a New Transaction

The following fields are used to define a menu item that starts a new transaction. For valid values, press PF1.

<b>Rule</b>	Type the name of the rule to be invoked when the user selects this item from the menu.
<b>Action New/Call/Menu</b>	Type N for NEW. This indicates that the menu item starts a new transaction when invoked.
<b>Search S/I/L</b>	Specify the level of library to start searching for the designated rule.
<b>Browse Y/N</b>	Set to Y to execute the rule in browse mode.
<b>Parm Y/N</b>	Specify whether (Y) the user can enter arguments for the rule.
<b>Prompt Obj/Parm/None</b>	If the <b>Parm Y/N</b> field is set to Y, specify the way in which the user is prompted. If you specify N and the user does not enter anything, the rule is passed a null for each of its arguments.
<b>Entry Length</b>	Specify a value for the maximum length of information the user can enter in the prompting field. The length of the prompting field depends on the argument(s) for the specified rule.

<b>Fill</b>	Type a character to be used as a fill character. The character appears repeatedly to the maximum length of the <b>Entry Length</b> field.
<b>Obj list or Menu</b>	<p>If the <b>Prompt</b> field is set to O, type the name of the object list. The object list appears in the <b>Obj list or Menu</b> field.</p> <p>The name you enter must be the same as the table name used when you defined the object list, using the <a href="#">DEFINE_OBJLIST</a> tool. You do not have to predefine the object list; it must be defined by the time you want to use <a href="#">DEFINE_MENU</a> or use the menu at login time. Refer to <i>TIBCO Object Service Broker Shareable Tools</i> for more information on the <a href="#">DEFINE_OBJLIST</a> tool and <a href="#">Appendix A, Displaying an Application, on page 115</a> for more information on the <a href="#">DEFINE_MENU</a> tool.</p>
<b>Usage</b>	Type a comment line. This line explains the use of the menu item and can be included in the help for your menu. Refer to <a href="#">Adding Help Information on page 112</a> for information on adding help to menus.
<b>Purpose</b>	Type the purpose of the menu item. You can include this field in the help for your menu. Refer to <a href="#">Adding Help Information on page 112</a> for information on adding help to menus.

Calling a Rule

The following fields are used to define a menu item that calls a TIBCO Object Service Broker rule. For valid values, press PF1. Some fields are not supported for a selection that calls a rule, for example, you cannot set the browse mode or search path.

<b>Rule</b>	Type a rule name. The rule is invoked when the user selects the item from the menu.
<b>Action New/Call/Menu</b>	Type C for CALL. This indicates that the menu item calls a TIBCO Object Service Broker rule. If you call a rule, you cannot use this menu as a login menu.
<b>Parm Y/N</b>	Type N. The rule called cannot have arguments.

<b>Prompt Obj/Parm/None</b>	Type N. The rule cannot have arguments.
<b>Usage</b>	Type a comment line. This line explains the use of the menu item and can be included in the help for your menu. Refer to <a href="#">Adding Help Information on page 112</a> for information on adding help to menus.
<b>Purpose</b>	Type the purpose of the menu item. You can include this field in the help for your menu. Refer to <a href="#">Adding Help Information on page 112</a> for information on adding help to menus.

## Calling Another Menu

The following fields are used to define a menu item that calls another menu. For valid values, press PF1. Some fields are not supported for a selection that calls another menu, for example, you cannot specify a library level.

<b>Action New/Call/Menu</b>	Type M for MENU. This indicates that the menu item calls another menu.
<b>Obj list or Menu</b>	Specify the name of the menu to appear. This menu must be defined using <code>DEFINE_MENU</code> . You do not have to predefine the referenced menu; it must be defined before the menu referencing it can be used. Refer to <a href="#">Appendix A, Displaying an Application, on page 115</a> for more information on the <code>DEFINE_MENU</code> tool.
<b>Usage</b>	Type a comment line. This line explains the use of the menu item and can be included in the help for your menu. Refer to <a href="#">Adding Help Information on page 112</a> for information on adding help to menus.
<b>Purpose</b>	Type the purpose of the menu item. You can include this field in the help for your menu. Refer to <a href="#">Adding Help Information on page 112</a> for information on adding help to menus.



## Chapter 10    **Maintaining Menus**

This chapter describes how to maintain Menus.

### Topics

---

- [Copying Menu Definitions, page 108](#)
- [Editing Menus, page 109](#)
- [Deleting a Menu Definition, page 110](#)
- [Adding Help Information, page 112](#)

## Copying Menu Definitions

---

### Creating Menu Definition Copies

You can only copy a menu from the Menu Definer screen. To do this, complete the following tasks:

1. From the EX execute rule option, type `DEFINE_MENU(menuname)`.  
*menuname* is the name of the menu you want to copy. The definition screen for that menu appears.
2. At the Define Menu field, type over the existing menu name with a menu name that does not already exist.
3. Modify the menu, if required.
4. Press PF3 to save the menu.

This menu (with the current characteristics) is saved under the new name and the existing menu remains in the TIBCO Object Service Broker system.

### Copying Standard Session Manager Menus

Before you can save a Standard Session Manager menu under a new name, you must change the names of the screen tables to tables that do not already exist.

See Also *TIBCO Object Service Broker Shareable Tools* for information on how to copy objects.

# Editing Menus

## PF Keys and Primary Commands

The following table lists the PF keys and primary commands available for editing a menu definition.

PF Key	Primary Command	Function
PF4	ADDLINE	Adds a line for data entry.
PF5	ADDTITLE	Adds a title line to the menu.
PF6	TEST	Saves and tests the definition. To test the functionality of the menu, the objects it uses (for example, rules, other menus, tables) must exist already if you want to select the menu options displayed and have them function as specified.
PF9	EDHELP	Edits the Help screen associated with the Screen Manager, Selection Manager, and Standard Session Manager menu styles.
PF16	DELLINE	Deletes a line of data entry.
PF17	DELTITLE	Deletes a title line from the menu.

See Also

*TIBCO Object Service Broker Getting Started* for information on standard primary commands.

## Deleting a Menu Definition

There are a number of methods that you can use to delete a menu definition. Refer to the following table to determine the appropriate method to use:

If the menu...	Use...
Was promoted to a target system	Promotion system for that location.
Was not promoted and is on your local node	Menu Definer or <a href="#">DELETE_DEFN</a> .
Was not promoted and is on a node remote to the node where you are presently working	<a href="#">DELETE_DEFN</a> .

### Considerations when Deleting a Menu

If a menu definition was promoted to another (target) system, you must submit a change request through the Promotion system (of the source system) to have the deletion extended to the target system. If you do not issue a change request to delete the definition, the following occurs:

- The menu exists on the target system and no rights are associated with it on the source system.
- If a new menu with the same name is created on the source system, the menu cannot be promoted to the target system because a menu with the same name already exists there.

### Deleting a Definition Using the Menu Definer

When you are within an existing definition in the Menu Definer you can delete a definition by doing one of the following:

- Press PF22.
- Use the **DELETE** command.

In either case you are prompted to confirm the deletion.



## Deleting a Definition Using a Shareable Tool

The shareable tool `DELETE_DEFN` is available to you to delete existing definitions. Using this tool, if the correct security access is set up, you can delete definitions across TIBCO Object Service Broker nodes or within your local node. `DELETE_DEFN` is called from within a rule.

### Example Rule

The following rule calls the `DELETE_DEFN` tool to delete the definition of the DEPARTMENTS menu from Node A. Because a menu definition is being deleted, values are not required for the arguments *library* or *environment*.

```
DELETE_MENUDEFN;  
-----  
-+-----  
_ CALL DELETE_DEFN('MENU', 'DEPARTMENTS', '', ' ', 'NODEA', | 1  
_ ' '); |  
_-----
```

- See Also
- *TIBCO Object Service Broker Managing Deployment* for information about change requests.
  - *TIBCO Object Service Broker Shareable Tools* for information about the tools.

## Adding Help Information

Help information can be added for Screen Manager, Selection Manager, and Standard Session Manager menu styles. Help information is stored in the same table used by the screen definer so you can create and modify help information using PF9. You can also use the primary command **EDHELP** from the Menu Definer or in the Screen Definer for the Selection Manager and Screen Manager menu styles.



You cannot add help to a Menu Manager menu.

### Adding Help Information

- To add help information from the Menu Definer, complete the following tasks:
1. Press PF9 or type the primary command EDHELP.  
A help screen appears that you can edit using the text editor (**TED**).
  2. Type the appropriate SCRIPT commands and help text.  
If information appears in the **Usage** and **Purpose** fields of the Menu Definer screen, you can embed the information in the Help screen using the **.table** command described below.

See Also *TIBCO Object Service Broker Shareable Tools* for information on the **SCRIPT** and **TED** tools.

### SCRIPT Commands

You can use the following two **SCRIPT** commands to create help:

.setup help	Mandatory
.table menu_items ( <i>menuname</i> ) title, usage, purpose	Optional

For the Screen Manager and the Selection Manager styles of menus, you can use the full range of **SCRIPT** commands to format the help text.

## Help Screen Restrictions

Do not embed the FCNKEYS table in the Help screen. The menus that you create all use the predefined keys 2, 3, and 12; therefore, you cannot define function keys for a menu.

### Example

The following two examples illustrate unformatted screen help and the formatted screen help the users see when they use PF1.

### Unformatted Screen Help

The following example illustrates unformatted screen help for the menu SEL\_EMPLOYEE:

---

```

Enter HELP text for screen SEL_EMPLOYEE
-----
_ .setup help
_ .p.This menu is used to enter weekly expenses, and create and
_ browse weekly status reports.
_ You can browse the following two types of status reports:
_ .ul.
_ .li.Open
_ .li.Closed
_ .eul.
_ .p.Place your cursor on the line command field beside the item you want to
_ access and press Enter.
```

```
PFKEYS: 12=QUIT 13=PRINT 3=SAVE 5=SCRIPT
```

---

Formatted Screen Help

The following example illustrates formatted screen help for the menu  
SEL\_EMPLOYEE:

Formatted Output	Scroll: P
This menu is used to enter weekly expenses, and create and browse weekly status reports. You can browse the following two types of status reports:	
-> Open	
-> Closed	
Place your cursor on the line command field beside the item you want to access and press Enter.	

## Appendix A **Displaying an Application**

This appendix describes how to display Screens and Menus and how to use the PROCESS\_FCNKEY Tool.

### Topics

---

- [Displaying a Screen, page 116](#)
- [Displaying a Menu, page 117](#)
- [Using the PROCESS\\_FCNKEY Tool, page 118](#)

## Displaying a Screen

---

There are three TIBCO Object Service Broker rules statements you can use to display a screen from a rule: the DISPLAY statement, the UNTIL...DISPLAY statement, and the DISPLAY & TRANSFERCALL statement.

### Example of the DISPLAY Statement

The following example illustrates the DISP\_SCREEN rule, which uses the DISPLAY statement to display the EMPLOYEE\_EXPENSE screen:

```

RULE EDITOR ==>                                SCROLL: P
DISP_SCREEN;
-
- -----
- -----+-----
- DISPLAY EMPLOYEE_EXPENSE;                      | 1
- -----
-
PFKEYS: 1=HELP 3=END 12=CANCEL 13=PRINT 14=EXPAND 2=DOCUMENT 22=DELETE
```

---

- See Also
- *TIBCO Object Service Broker Programming in Rules* for information about DISPLAY, UNTIL...DISPLAY, and DISPLAY & TRANSFERCALL statements.
  - The Screen Tools section of *TIBCO Object Service Broker Shareable Tools* for information on displaying a screen

## Displaying a Menu

To display a menu from within an application, you can use the tool `DISPALY_MENU` tool. If the menu has sub-menus embedded within it, these menus are also called.

## Example of the DISPLAY\_MENU Tool

The following example illustrates the `DISPLAY_SCR_EMP` rule, which uses the `DISPLAY_MENU` tool to display the `SCR_EMPLOYEE` menu:

```

RULE_EDITOR ==>                                SCROLL: P
DISPLAY_SCR_EMP;

- -----+-----
-                                     | 1
- CALL DISPLAY_MENU( 'SCR_EMPLOYEE' );
- -----
```

PFKEYS: 1=HELP 3=END 12=CANCEL 13=PRINT 14=EXPAND 2=DOCUMENT 22=DELETE

## Using the PROCESS\_FCNKEY Tool

If you specify any function keys, you can use the [PROCESS\\_FCNKEY](#) tool in combination with any of the display statements to create a loop that automatically processes the user's PF key selections. For example, the following two rules show how the [PROCESS\\_FCNKEY](#) tool and the `DISPLAY & TRANSFERCALL` statement could be combined to provide pseudo-conversational processing of PF keys.

### Examples

The `PROCESS_EXPENSES` rule is called first. The first two lines initialize the scroll amount field. The third line displays the `EMPLOYEE_EXPENSE` screen and transfercalls to the `EXPENSE_LOOP` rule.

```
RULE EDITOR ===>                                SCROLL: P
PROCESS_EXPENSES;

-
- -----
-                                     +-----
- COMP_TITLE.SCROLL_AMT = 'P';                                | 1
- INSERT COMP_TITLE('EMPLOYEE_EXPENSE');                      | 2
- DISPLAY EMPLOYEE_EXPENSE & TRANSFERCALL EXPENSE_LOOP;       | 3
- -----
-

PFKEYS: 1=HELP 3=END 12=CANCEL 13=PRINT 14=EXPAND 2=DOCUMENT 22=DELETE
```



EXPENSE\_LOOP Rule

The first line of the EXPENSE\_LOOP rule uses the [PROCESS\\_FCNKEY](#) tool to handle any PF key selection the user makes. The second statement redisplay the screen and does a recursive transercall back to this same rule to process the next user PF key use. One of the function keys must signal the EXIT\_DISPLAY exception. This causes the loop to complete.

```

RULE EDITOR ==>
EXPENSE_LOOP;
-----
CALL PROCESS_FCNKEY('EMPLOYEE_EXPENSE');
DISPLAY EMPLOYEE_EXPENSE & TRANSFERCALL EXPENSE_LOOP;
ON EXIT_DISPLAY:
-----
-----+-----
| 1
| 2
-----
PFKEYS: 1=HELP 3=END 12=CANCEL 13=PRINT 14=EXPAND 2=DOCUMENT 22=DELETE
-----
```



## Appendix B **Display Masks**

This appendix describes how to use Display Masks.

### Topics

---

- [Display Masks, page 122](#)

## Display Masks

---

Screen fields with semantic type C (count), D (date), I (identifier), or Q (quantity) and fields that have syntax P (packed) or B (binary) can have display masks applied to them. You can apply two kinds of display masks:

- Numeric
- Date

### Numeric Display Masks

Numeric display masks can be applied to a field with a semantic type C (count), I (identifier), or Q (quantity) and syntax B (binary) or P (packed). You use a numeric display mask to print numeric fields according to common conventions, for example, 1,000 or \$1,000.00. For example, a field defined as ZZZZ9V.99 converts to 10000.00.

### Display Mask Entries

For numeric fields, you can make the following display mask entries:

- Type in any string consisting of characters to be printed, for example, dollar signs (\$) and commas (,).
- Type in digit placeholders. The digit placeholders reserve display spaces for the numeric digits. The digit placeholders you define must be sufficient to hold the largest value that is processed for your field. You can define a maximum of 15 digit placeholders.
- If negative data can appear, include the negative sign with the mask. Examples of acceptable masks are -ZZZZZ9V.99 and \$ZZZZ9V.99.
- The only digit allowed in the display mask specification is 9.
- Only one decimal separator display character is allowed in the display mask basic string.
- A non-numeric, non-digit character is required for a decimal separator display character (for example, 999V.9 could print as 123.4).

- If a user inserts values greater than a display mask allows, the following results occur:

Method of Insertion	Results
Typed into the left of the decimal point.	The OVERFLOW exception is raised.
Typed into the right of the decimal portion.	The value is truncated to a point on the display, based on the decimal length.
Via a rule.	The data is presented on the display as asterisks (*).

Characteristics of Digit Placeholders

You can use the following alphanumeric characters to represent digit placeholders. Each character has a specific representation:

9	Leading zero and null appear as 0.
Z	Leading zero and null appear as blank.
N	Leading zero and null appear as null.
*	Null and 0 print as asterisks (*).
V	Decimal placeholder.

Date Display Masks

The following table shows valid date display mask components for fields with semantic type D (date) and syntax B (binary).

Format Code	Meaning	Example
W	One or two digit week # (of year), with no leading 0.	1 or 25
WW	Two digit week # (of year).	01
WWW	Abbreviated weekday.	Mon
WWWW	Full weekday.	Monday

Format Code	Meaning	Example
M	Numeric month, with no leading 0 (1 or 2 digits).	3 or 10
MM	Numeric month (2 digits).	02
MMM	Abbreviated month.	Mar
MMMM	Full month.	March
D	Day in month, with no leading 0 (1 or 2 digits).	5 or 14
DD	Day in month (2 digits).	02
DDD	Day in year (3 digits).	074
YY	Last two digits in a year.	00
YYYY	Full year.	2000
QQ	Two character quarter.	2Q
JD	Julian date.	00.074
CC	Two digit century.	20



Note the following considerations when applying display masks:

- A separator character can be any one of the following:  
/ \ ; : , . \* - blank
- If no date format is specified, the installation default date format is used.
- A week is defined to begin on a Monday and end on the following Sunday. However, January 1st always begins week one, regardless of where it falls in the week, and week two starts on the following Monday.
- D B 4 fields can handle a minimum date of 0001/01/01 and a maximum date of 9999/12/31.
- An input in YY/MM/DD of 01/01/01 results in a date from 1901. To avoid confusion, you should specify the century in your dates.
- Do not make display masks longer than the field length. If the display mask is longer than the field length, the screen server takes the formatted value and displays it in left to right order, in the desired display length. Extra formatted digits are dropped.
- You can specify just a portion of a date field within your mask (for example, entering only MMMM displays the month). Partial date occurrences cannot be accessed using a GET or FORALL statement, as the data cannot be interpreted as a complete date. At least the year portion of a date must be present in the mask to make it accessible to these statements.

**See Also** *TIBCO Object Service Broker Shareable Tools* for information on the [\\$PIC](#) tool (the display mask specifications you can make are based on this tool) and the Date tools for information on date display masks.





## Appendix C    **Sample Screen Definitions for Menus**

This appendix provides the definitions for sample Screens.

### Topics

---

- [Definition for the WEEKLY\\_REPORT Screen, page 128](#)
- [Definition for the WEEKLY\\_REP Screen, page 133](#)
- [Definition for SCR\\_EMPLOYEE Menu, page 138](#)

## Definition for the WEEKLY\_REPORT Screen

The following sections provide sample definitions for the WEEKLY\_REPORT screen and its screen tables used for the menu in shown [Chapter 8, Creating a Selection Manager Menu](#):

- TITLE
- SELECTION
- PFKEY\_SPECS
- SESSMGR\_MSG

### WEEKLY\_REPORT Screen

The following example illustrates the definition for the WEEKLY\_REPORT screen:

COMMAND ==>				DEFINE SCREEN: WEEKLY_REPORT				Unit: CSS			
PFkey Assignments				Scroll Amount Entry				Default Cursor Position			
-----				-----				-----			
Up: 7		Down: 8		Table:				Table: SELECTION			
Left: 10		Right: 11		Field:				Field: EXPENSES			
Validation Exit: 12											
Help: 1		Refresh: 24									
SCREEN TABLES:		Origin		Max				Title		Fix	
Name		Row Col		occ Scroll		Validation Rule		Rows		Col	
-----		----		----		-----		----		----	
_ TITLE		1 1		1 N				0		0	
_ SELECTION		5 1		1 N				0		0	
_ PFKEY_SPECS		-2 1		1 N				0		0	
_ SESSMGR_MSG		-1 1		1 N				0		0	
_											
_											
_											
_											
PFKEYS: 6=PAINT 16=EXCLD 21=DISPLAY 18=SPEC_FCNKEYS 19=SCR_HELP 13=PRINT											

TITLE Screen Table

The following example illustrates the definition for the TITLE screen table:

SCREEN PAINTER COMMAND ==>

Scroll: P

.....1.....2.....3.....4.....5.....6.....7.....

-99999999999

SOFTWARE DEVELOPMENT CORPORATION

UNIT

-AAAAAA

Table: TITLE

Unit: CSS

ROW	COL	FIELD NAME	Type	Syn	Len	Dec	Just	Fill	Prot	Show	Rqd	Hi	Skip	Null
-----			-	-	----	----	-	-	-	-	-	-	-	-
1	1	DATE	D	B	11	0	L	.	Y	Y	N	N	Y	Y
1	68	USERID	S	C	6	0	L	.	Y	Y	N	N	Y	Y

PFKEYS: 6=+FLD 18=-FLD 4=+LINE 5=CUT 19=FLD\_HELP 17=PASTE 16=-LINE 13=PRINT

SELECTION Screen Table

The following example illustrates the definition for the SELECTION screen table:

SCREEN PAINTER COMMAND ==> Scroll: P														
.....1.....2.....3.....4.....5.....6.....7.....														
-A Weekly Expenses														
-A Weekly Status Report -A open -A closed														
Press ENTER beside the option desired														
Table: SELECTION Unit: CSS														
ROW	COL	FIELD NAME	Type	Syn	Len	Dec	Just	Fill	Prot	Show	Rqd	Hi	Skip	Null
---	---	-----	-	-	---	---	-	-	-	-	-	-	-	-
1	22	EXPENSES	S	C	1	0	L	-	N	Y	N	N	Y	Y
3	22	STATUS	S	C	1	0	L	-	N	Y	N	N	Y	Y
3	50	OPEN	S	C	1	0	L	-	N	Y	N	N	Y	Y
3	60	CLOSED	S	C	1	0	L	-	N	Y	N	N	Y	Y
PFKEYS: 6=+FLD 18=-FLD 4=+LINE 5=CUT 19=FLD_HELP 17=PASTE 16=-LINE 13=PRINT														

## PFKEY\_SPECS Screen Table

The following example illustrates the definition for the PFKEY\_SPECS screen table:

```
SCREEN PAINTER COMMAND ==>                                Scroll: P
....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....
~AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

Table: PFKEY\_SPECS

Unit: UTIL

ROW	COL	FIELD NAME	Type	Syn	Len	Dec	Just	Fill	Prot	Show	Rqd	Hi	Skip	Null
---	---	-----	-	-	---	---	-	-	-	-	-	-	-	-
1	1	PFKEYS	S	C	78	0	L		Y	Y	N	Y	Y	Y

```
PFKEYS: 6=+FLD 18=-FLD 4=+LINE 5=CUT 19=FLD_HELP 17=PASTE 16=-LINE 13=PRINT
```

## SESSMGR\_MSG Screen Table

The following example illustrates the definition for the SESSMGR\_MSG screen table:

SCREEN PAINTER COMMAND ==>

Scroll: P

.....1.....2.....3.....4.....5.....6.....7.....

-AA

Table: SESSMGR\_MSG

Unit: SMG

ROW	COL	FIELD NAME	Type	Syn	Len	Dec	Just	Fill	Prot	Show	Rqd	Hi	Skip	Null
---	---	-----	-	-	----	---	-	-	-	-	-	-	-	-
1	1	MSG		V	78	0	L		Y	Y	N	N	N	N

PFKEYS: 6=+FLD 18=-FLD 4=+LINE 5=CUT 19=FLD\_HELP 17=PASTE 16=-LINE 13=PRINT

# Definition for the WEEKLY\_REP Screen

The following sections provide sample definitions for the WEEKLY\_REP screen and its screen tables used in the menu definition in [Chapter 7, Creating a Screen Manager Menu](#):

- TITLE\_1
- FIELD
- PRESS
- PFKEY\_SPECS
- SESSMGR\_MSG

## WEEKLY\_REP Screen

The following example illustrates the definition for the WEEKLY\_REP screen:

DEFINE SCREEN: WEEKLY_REP					Unit: CSS			
COMMAND ==>								
PFkey Assignments			Scroll Amount Entry			Default Cursor Position		
-----			-----			-----		
Up: 7	Down: 8	Table:	Table: FIELD					
Left: 10	Right: 11	Field:	Field: VALUE					
Validation Exit: 12								
Help: 1 Refresh: 24								
SCREEN TABLES:		Origin	Max	Scroll	Validation Rule	Title	Fix	Last
Name		Row Col	occ			Rows	Col	Row Col
-----		---	---	---	-----	---	---	---
_ TITLE_1		1 1	1	N		0	0	
_ FIELD		5 1	*	Y		0	0	
_ PRESS		-15 1	1	N		0	0	
_ PFKEY_SPECS		-2 1	1	N		0	0	
_ SESSMGR_MSG		-1 1	1	N		0	0	
_								
_								
_								
PFKEYS: 6=PAINT 16=EXCLD 21=DISPLAY 18=SPEC_FCNKEYS 19=SCR_HELP 13=PRINT								

TITLE\_1 Screen Table

The following example illustrates the definition for the TITLE\_1 screen table:

SCREEN PAINTER COMMAND ==>

Scroll: P

.....1.....2.....3.....4.....5.....6.....7.....

SOFTWARE DEVELOPMENT CORPORATION

-999999999999

Table: TITLE\_1

Unit: CSS

ROW	COL	FIELD NAME	Type	Syn	Len	Dec	Just	Fill	Prot	Show	Rqd	Hi	Skip	Null
---	---	-----	-	-	----	---	-	-	-	-	-	-	-	-
2	30	DATE	D	B	11	0	L	.	Y	Y	N	N	Y	Y

PFKEYS: 6=+FLD 18=-FLD 4=+LINE 5=CUT 19=FLD\_HELP 17=PASTE 16=-LINE 13=PRINT



FIELD Screen Table

The following example illustrates the definition for the FIELD screen table:

SCREEN PAINTER COMMAND ==>

Scroll: P

.....1.....2.....3.....4.....5.....6.....7.....

-A -AAAAAAAAAAAAAAAAAAAAAAAAAAAA

Table: FIELD

Unit: CSS

ROW	COL	FIELD NAME	Type	Syn	Len	Dec	Just	Fill	Prot	Show	Rqd	Hi	Skip	Null
---	---	-----	-	-	----	---	-	-	-	-	-	-	-	-
1	25	VALUE	S	V	1	0	L	-	N	Y	N	N	Y	Y
1	28	TITLE	S	V	25	0	L		Y	Y	N	N	Y	Y

PFKEYS: 6=+FLD 18=-FLD 4=+LINE 5=CUT 19=FLD\_HELP 17=PASTE 16=-LINE 13=PRINT

## PRESS Screen Table

The following example illustrates the definition for the PRESS screen table:

```
SCREEN PAINTER COMMAND ==> Scroll: P
...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...
      Press ENTER beside the option desired
```

Table: PRESS			Unit: CSS											
ROW	COL	FIELD NAME	Type	Syn	Len	Dec	Just	Fill	Prot	Show	Rqd	Hi	Skip	Null

PFKEYS: 6=+FLD 18=-FLD 4=+LINE 5=CUT 19=FLD\_HELP 17=PASTE 16=-LINE 13=PRINT

## PFKEY\_SPECS Screen Table

The following example illustrates the definition for the PFKEY\_SPECS screen table:

[illegible]

Table: PFKEY_SPECS			Unit: UTIL											
ROW	COL	FIELD NAME	Type	Syn	Len	Dec	Just	Fill	Prot	Show	Rqd	Hi	Skip	Null
---	---	-----	-	-	----	---	-	-	-	-	-	-	-	-
1	1	PFKEYS	S	C	78	0	L		Y	Y	N	Y	Y	Y

PFKEYS: 6=+FLD 18=-FLD 4=+LINE 5=CUT 19=FLD\_HELP 17=PASTE 16=-LINE 13=PRINT

---

SESSMGR\_MSG Screen Table

The following example illustrates the definition for the SESSMGR\_MSG screen table:

---

```
SCREEN PAINTER COMMAND ==>                                Scroll: P
....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....
~AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

Table: SESSMGR_MSG			Unit: SMG											
ROW	COL	FIELD NAME	Type	Syn	Len	Dec	Just	Fill	Prot	Show	Rqd	Hi	Skip	Null
----	----	-----	-	-	----	----	-	-	-	-	-	-	-	-
1	1	MSG		V	78	0	L		Y	Y	N	N	N	N

PFKEYS: 6=+FLD 18=-FLD 4=+LINE 5=CUT 19=FLD\_HELP 17=PASTE 16=-LINE 13=PRINT

---

## Definition for SCR\_EMPLOYEE Menu

---

The following illustrates the full definition of the SCR\_EMPLOYEE menu, which spans across four screens shown in [Chapter 7, Creating a Screen Manager Menu](#). To scroll to the right, press PF11 and to scroll to the left, press PF10.

### Screen 1

The following example illustrates the first screen of the SCR\_EMPLOYEE menu definition:

---

Define Menu: SCR_EMPLOYEE	Unit: EXMPL
Command ==>	
	Screen to be used: WEEKLY_REP_S
Scrolling table containing fields "TITLE" and "VALUE": FIELD_ST	
Optional table with field named "DATE": DIANE_SCRDTBL	
-----	
Menu Items	
Time & Ok: Y	Action
Title	Search Browse
-----	New/Call/Menu S/I/L Y/N
Create an expense report	M
Create a status report	WEEKLY_STA_R C L N

PFKEYS: 3=SAVE 12=CANCEL 4=+LINE 16=-LINE 22=DEL 6=TEST 9=EDHELP 2=DOC
Left edge of Window

---

Screen 2

The following example illustrates the second screen of the SCR\_EMPLOYEE menu definition. To view this portion of the screen, press PF11.

Define Menu: SCR\_EMPLOYEE

Unit: EXMPL

Command ==>

Screen to be used: WEEKLY\_REP\_S

Scrolling table containing fields "TITLE" and "VALUE": FIELD\_ST

Optional table with field named "DATE": DIANE\_SCRTBL

-----

Time & Ok: Y	Parm	Prompt	Entry	Obj list
Title	Y/N	Obj/Parm/None	Length Fill	or Menu
-----	-	-	--	- -----
Create an expense report	Y	O	0	SEL_EMPLOYEE
Create a status report	N	N	1	

-----

PFKEYS: 3=SAVE 12=CANCEL 4=+LINE 16=-LINE 22=DEL 6=TEST 9=EDHELP 2=DOC

Screen 3

The following example illustrates the third screen of the SCR\_EMPLOYEE menu definition. To view this portion of the screen, press PF11.

```
Define Menu: SCR_EMPLOYEE                               Unit: EXMPL
  Command ==>
                                Screen to be used: WEEKLY_REP_S
Scrolling table containing fields "TITLE" and "VALUE": FIELD_ST
                                Optional table with field named "DATE": DIANE_SCRTBL
-----

Time & Ok: Y      t
      Title      -----      Usage
      -----
Create an expense report  EE
Create a status report
```

PFKEYS: 3=SAVE 12=CANCEL 4=+LINE 16=-LINE 22=DEL 6=TEST 9=EDHELP 2=DOC

Screen 4

The following example illustrates the fourth screen of the SCR\_EMPLOYEE menu definition. To view this portion of the screen, press PF11.

---

Define Menu: SCR\_EMPLOYEE

Unit: EXMPL

Command ==>

Screen to be used: WEEKLY\_REP\_S

Scrolling table containing fields "TITLE" and "VALUE": FIELD\_ST

Optional table with field named "DATE": DIANE\_SCRTBL

-----

Time & Ok: Y	
Title	Purpose
-----	-----
Create an expense report	
Create a status report	

PFKEYS: 3=SAVE 12=CANCEL 4=+LINE 16=-LINE 22=DEL 6=TEST 9=EDHELP 2=DOC

Right edge of Window

---





# Index

## Symbols

@COLOURS table 60  
 @SCREENCOLOURS table 61  
 @TMP\_FCNKEYS table 18  
 \$MODIFIED field 63  
 \$OCCNUM field 64

## A

abbreviations, displaying menu items 102  
 accessing  
   data 42  
   Menu Definer tool 71, 81, 92, 100  
   Table Definer 8  
 Add Field key 40  
 Add Line key 49, 109  
 Add Title key 109  
 adding  
   background color 46, 46  
   brightness to screen fields 42  
   colors 60  
   decimals to screen fields 42  
   field help 56  
   foreground color 46  
   functionality to menus  
     Menu Manager menu 74  
     Screen Manager menu 83  
     Selection Manager menu 94  
     Standard Session Manager menu 103  
   lines 49  
   literal text 31  
   menu help 112  
   menu items 73, 82, 93, 102  
   screen fields 49  
   screen help 54  
 ADDLINE primary command 109

ADDTITLE primary command 109  
 APPEND primary command 36  
 appending fields 37  
 applications, displaying  
   menus within 117  
   screens within 116  
 appointment calendar, including in menus 67  
 arguments  
   enabling users to enter  
     Menu Manager menu 74, 75  
     Screen Manager menu 83  
     Selection Manager menu 94  
     Standard Session Manager menu 103, 104  
   list prompts, Standard Session Manager menu 103  
 arrows, displaying menu items 101  
 assigning PF key values 15

## B

background color, adding 46  
 benefits  
   default fields 35  
   new fields 35  
   predefined fields 34  
 brightness, adding to screen fields 42  
 browse flag  
   and menus 68  
   excluding 101  
   including 101  
 browse mode  
   Menu Manager menu 74  
   Screen Manager menu 83  
   Selection Manager menu 94  
   Standard Session Manager menu 103

## C

- calendar
  - excluding 100
  - including 100
- calling
  - menus
    - Menu Manager menu 76
    - Screen Manager menu 85
    - Selection Manager menu 96
    - Standard Session Manager menu 105
  - rules
    - Menu Manager menu 75
    - Screen Manager menu 84
    - Selection Manager menu 95
    - Standard Session Manager menu 104
- Cancel key 16
- centering title lines 100
- Change Literal Defaults key 32
- choosing menu styles 67
- colors
  - adding 60
  - supported for display devices 61
- columns
  - fixed 13
  - in menus 67
  - last 14
  - single, in menus 67
- command history
  - area, and menus 67
  - default length 102
  - displaying 102
- command lines
  - displaying 102
  - in menus 67
- comment lines
  - Menu Manager menu 75, 75, 76
  - Screen Manager menu 84, 85, 85
  - Selection Manager menu 95, 96, 96
  - Standard Session Manager menu 104, 105, 105
- controlling presentation for screen fields 44
- Copy Line key 49
- COPY primary command 36

- copying
  - fields 36
  - lines 49
  - literal text 49
  - menu definitions 108
- creating
  - literal text 31
  - Menu Manager menu 69
  - new fields 41
  - Screen Manager menu 77
  - Selection Manager menu 87
  - Standard Session Manager menu 97
- creators, user IDs of 100
- current
  - date, displaying
    - Menu Manager menu 72
    - Screen Manager menu 81
    - Selection Manager menu 92
    - Standard Session Manager menu 100
  - position of
    - cursor 64
    - occurrences 64
  - time, displaying
    - Menu Manager menu 73
    - Screen Manager menu 82
    - Selection Manager menu 93
    - Standard Session Manager menu 102, 102
- cursor position
  - current 64
  - default 17
- customer support xviii
- Cut Field key 49
- cutting
  - literal text 49
  - screen fields 49

## D

- D
  - (delete) line command 51, 51

- data
  - accessing 42
  - justifying 42
  - use 44
- data fields
  - defining 34
  - methods for defining 34
- date field, including in menus 68
- dates
  - current, displaying
    - Menu Manager menu 72
    - Screen Manager menu 81
    - Selection Manager menu 92
    - Standard Session Manager menu 101
  - display masks 123
- default
  - command history length 102
  - cursor position, defining 17
  - fields
    - benefits 35
    - using 40
  - scroll amount, defining 16
- DEFINE SCREEN field 10
- DEFINE\_MENU tool 76, 85, 96, 105
- DEFINE\_OBJLIST tool 75, 84, 95, 104
- defining
  - command area 102
  - data fields 34
  - default
    - cursor position 17
    - scroll amount 16
  - head area 101
  - menu area 101
  - optional screen table attributes 13
  - PF keys 18
  - physical layout
    - Menu Manager menu 71
    - Screen Manager menu 79, 81
    - Selection Manager menu 90, 92
    - Standard Session Manager 99
  - required screen table attributes 11
  - screen tables 30
  - screens 9
  - session menu area 100
- definitions
  - deleting 20, 20, 110, 110
  - expanding 47
  - literal text 31
  - menus 3
  - screen fields 26
  - screen tables 2
  - screens 2
- delete (D) line command 51, 51
- Delete Field key 50
- Delete Line key 50, 109
- DELETE primary command 20, 110
- Delete Screen Table key 51
- Delete Title key 109
- DELETE\_DEFN tool 21, 111
- deleting
  - definitions
    - available tools 21, 111
    - considerations when deleting 20, 110
    - using Screen Definer 20, 110
  - lines 50
  - screen fields 50
  - screen tables 50
- DELLINE primary command 109
- DELTITLE primary command 109
- description, menu items
  - Menu Manager menu 73
  - Screen Manager menu 82
  - Standard Session Manager menu 102
- determining data use 44
- DISPLAY & TRANSFERCALL statement 116
- display attributes
  - for literal text 32
  - setting for literal text 32
- display masks
  - for dates 123
  - numeric 122
- Display Screen key 9
- DISPLAY statement 116
- DISPLAY\_MENU tool 117
- displaying
  - command history 102
  - command lines 102
  - current date
    - Menu Manager menu 72

- Screen Manager menu [81](#)
- Selection Manager menu [92](#)
- Standard Session Manager menu [101](#)
- current time
  - Menu Manager menu [73](#)
  - Screen Manager menu [82](#)
  - Selection Manager menu [93](#)
  - Standard Session Manager menu [101, 102, 102](#)
- menu items
  - abbreviations [102](#)
  - arrows [101](#)
- menus within applications [117](#)
- OK
  - Menu Manager menu [73](#)
  - Screen Manager menu [82](#)
  - Selection Manager menu [93](#)
  - Standard Session Manager menu [102](#)
- partial screen fields [46](#)
- screens within applications [116](#)
- user IDs
  - Menu Manager menu [72](#)
  - Selection Manager menu [92](#)

## E

- EDHELP primary command [109](#)
- Edit Help key [109](#)
- editing
  - menus [109](#)
  - screen fields [49](#)
  - screen tables [49](#)
- emphasizing fields [46](#)
- empty occurrences [14](#)
- enabling users to enter arguments
  - Menu Manager menu [74, 75](#)
  - Screen Manager menu [83](#)
  - Selection Manager menu [94](#)
  - Standard Session Manager menu [103, 104](#)
- entering PF key, scrolling, and cursor information [15](#)
- exception, VALIDATEFAIL [4](#)
- Exclude Table key [14](#)

- excluding
  - browse flag [101](#)
  - calendar [100](#)
  - library names [101](#)
  - screen tables [14](#)
  - test flag [101](#)
- Expand key [47](#)
- expanding
  - definitions [47](#)
  - literal text [47](#)
  - screen field definitions [47](#)

## F

- field definition area [29](#)
- Field Help key [56](#)
- field information section [29](#)
- fields
  - adding help [56](#)
  - definitions from tables [34](#)
  - filling with nulls [43](#)
  - making required [43](#)
  - names for scroll fields [16, 17](#)
  - protecting [43](#)
  - showing content [43](#)
  - skipping [43](#)
  - underlining [46](#)
- Fields key [38](#)
- fill characters
  - Menu Manager menu [75](#)
  - Screen Manager menu [84](#)
  - Selection Manager menu [95](#)
  - Standard Session Manager menu [104](#)
  - using [45](#)
- filling fields with nulls [43](#)
- first
  - column of screen tables [12](#)
  - row of screen tables [11](#)
- fixed
  - columns [13](#)
- foreground color, adding [46](#)
- formatting fields [45](#)
- function key attributes, specifying [18](#)

## G

global fields  
     defined 34  
     selecting 38  
     using 38

## H

header information section 29  
 headers 100  
 help  
     levels available 54  
     scripting text 54, 56, 56

## I

image area 29  
 including  
     browse flag 101  
     calendar 100  
     library names 101  
     test flag 101  
 incorporating shareable design elements in menus 67  
 indicating, scrolling direction 45  
 invoking  
     Menu Definer tool 71, 81, 92, 100  
     Screen Table Painter 27

## J

justifying data 42

## L

last  
     columns 14  
     rows 13  
 layout  
     Screen Definer 8  
     Screen Table Painter 28  
 length of prompt response  
     Menu Manager menu 74  
     Screen Manager menu 83  
     Selection Manager menu 94  
     Standard Session Manager menu 103  
 level of help 54  
 libraries  
     including names on menus 68  
     names  
         excluding 101  
         including 101  
     search  
         Menu Manager menu 74  
         Screen Manager menu 83  
         Selection Manager menu 94  
         Standard Session Manager menu 103  
 limits for screen fields 26  
 line commands, delete (D) 51, 51  
 lines  
     adding 49  
     copying 49  
     deleting 50  
 list of supported colors 60  
 literal text  
     adding 31  
     copying 49  
     creating 31  
     cutting 49  
     defined 31  
     display attributes 32  
     expanding 47  
     pasting 49  
     setting display attributes 32

## M

- maintaining menus [107](#)
- maximum
  - number of row occurrences [12](#)
  - occurrences in screen tables [11](#)
  - screen table width [12](#)
- Menu Definer tool, invoking [71, 81, 92, 100](#)
- menu items
  - abbreviations, displaying [102](#)
  - arrows, displaying [101](#)
  - description [73, 82, 102](#)
- Menu Manager menu
  - adding
    - functionality [74](#)
    - menu items [73](#)
  - creating [69](#)
  - defining physical layout [71](#)
  - description of menu items [73](#)
  - overview [70](#)
  - specifying titles [72](#)
  - styles [66](#)
- menus
  - adding help [112](#)
  - and predefined screens [67](#)
  - appointment calendar [67](#)
  - browse flag [68](#)
  - calling
    - Menu Manager menu [76](#)
    - Screen Manager menu [85](#)
    - Selection Manager menu [96](#)
    - Standard Session Manager menu [105](#)
  - command history area [67](#)
  - command lines [67](#)
  - defined [3, 66](#)
  - including
    - date field [68](#)
    - library names [68](#)
    - time field [68](#)
  - maintaining [107](#)
  - number of columns [67](#)
  - shareable design elements [67](#)
  - styles
    - choosing [67](#)
    - Menu Manager menu [66](#)
    - Screen Manager menu [66](#)
    - Selection Manager menu [66](#)
    - Standard Session Manager menu [66](#)
  - test flag [68](#)
  - to be invoked
    - Menu Manager menu [76](#)
    - Screen Manager menu [85](#)
    - Selection Manager menu [96](#)
    - Standard Session Manager menu [105](#)
  - using TIBCO Object Service Broker screens [67](#)
  - with single column [67](#)
- methods for defining data fields [34](#)
- modes
  - browse
    - Menu Manager menu [74](#)
    - Screen Manager menu [83](#)
    - Selection Manager menu [94](#)
    - Standard Session Manager menu [103](#)
  - update
    - Menu Manager menu [74](#)
    - Screen Manager menu [83](#)
    - Selection Manager menu [94](#)
    - Standard Session Manager menu [103](#)
- modified occurrences [63](#)
- modifying
  - names of menus, Standard Session Manager menu [100](#)
  - screen table names, Standard Session Manager menu [101, 101, 102](#)

## N

- names
  - menus, modifying [100](#)
- object lists
  - Menu Manager menu [75](#)
  - Screen Manager menu [84](#)
  - Selection Manager menu [95](#)
  - Standard Session Manager menu [104, 104](#)
- of menus to be invoked
  - Menu Manager menu [76](#)
  - Screen Manager menu [85](#)
  - Selection Manager menu [96](#)

- Standard Session Manager menu 105
- of screen fields, Selection Manager menu 93
- of screen tables, modifying 101, 102
- screen tables, modifying 101
- naming screen fields 41
- new
  - fields
    - benefits 35
    - creating 41
  - names, saving
    - screen definitions under 10
    - screen table definitions under 11
  - transactions, starting
    - Menu Manager menu 74
    - Screen Manager menu 83
    - Selection Manager menu 94
    - Standard Session Manager menu 103
- NONAMES option 36, 37
- numeric display masks 122

## O

- object list names
  - Menu Manager menu 75
  - Screen Manager menu 84
  - Selection Manager menu 95
  - Standard Session Manager menu 104
- occurrences
  - current position of 64
  - modified 63
  - partial 14
- OK, displaying
  - Menu Manager menu 73
  - Screen Manager menu 82
  - Selection Manager menu 93
  - Standard Session Manager menu 102

## P

- Paint Screen Table key 27

- partial
  - display of screen fields 46
  - occurrences 14
- Paste key 50
- pasting
  - literal text 49
  - screen fields 49
- PF Key, Scrolling, and Cursor Information segment 15
- PF keys
  - defining 18
- Menu Definer
  - PF16 (delete line) 109
  - PF17 (delete title) 109
  - PF4 (add line) 109
  - PF5 (add title) 109
  - PF6 (test) 109
  - PF9 (edit help) 109
- re-assigning 15
- Screen Definer
  - PF12 (cancel) 16
  - PF16 (exclude table) 14
  - PF18 (specify function keys) 18
  - PF19 (screen help) 54
  - PF21 (display screen) 9
  - PF24 (refresh) 15
  - PF6 (paint screen table) 27
- Screen Table Painter
  - PF14 (fields) 38
  - PF15 (expand) 47
  - PF16 (delete line) 50
  - PF17 (paste) 50
  - PF18 (delete field) 50
  - PF19 (field help) 56
  - PF20 (copy line) 49
  - PF22 (delete screen table) 51
  - PF23 (reference screen table) 51
  - PF4 (add Line) 49
  - PF5 (cut field) 49
  - PF6 (add field) 40
  - PF9 (change literal defaults) 32
- Phase One validation 4
- Phase Two validation 4

physical layout, defining

- Menu Manager menu [71](#)
- Screen Manager menu [79](#)
- Selection Manager menu [90](#)
- Standard Session Manager [99](#)

predefined

fields

- benefits [34](#)
- using [36](#)

screens

- Screen Manager menu [78, 81](#)
- Selection Manager menu [88, 92](#)

predefining screens

- Screen Manager menu [80](#)
- Selection Manager menu [91](#)

presentation, controlling for screen fields [44](#)

primary command line, displaying [102](#)

primary commands

- ADDLINE [109](#)
- ADDTITLE [109](#)
- APPEND [36](#)
- COPY [36](#)
- DELETE [20, 110](#)
- DELLINE [109](#)
- DELTITLE [109](#)
- EDHELP [109](#)
- TEST [109](#)

PROCESS\_FCNKEY tool, using [118](#)

prompting

- for selection values [44](#)
- supplying [44](#)

prompts

response, length of

- Menu Manager menu [74](#)
- Screen Manager menu [83](#)
- Selection Manager menu [94](#)
- Standard Session Manager menu [103](#)

types

- Menu Manager menu [74, 75](#)
- Screen Manager menu [83](#)
- Selection Manager menu [94](#)
- Standard Session Manager menu [103](#)

protecting fields [43](#)

purposes of menu items

- Menu Manager menu [75, 76, 76](#)
- Screen Manager menu [84, 85, 85](#)
- Selection Manager menu [95, 96](#)
- Standard Session Manager menu [104, 105, 105](#)

purposes of menus item

- Selection Manager menu [96](#)

## R

re-assigning PF keys [15](#)

Reference Screen Table key [51](#)

reference tables, using [44](#)

Refresh key [15](#)

requiring values for fields [43](#)

reversing video [46](#)

rows, last [13](#)

rules

executed during screen display [13](#)

to be invoked

- Menu Manager menu [74, 75](#)
- Screen Manager menu [83, 84](#)
- Selection Manager menu [94, 95](#)
- Standard Session Manager menu [103, 104](#)

## S

saving

- screen definitions under new names [10](#)
- screen table definitions under new name [11](#)

Screen Definer

- deleting definitions [20, 110](#)
- screen layout [8](#)



- screen fields 41, 46
  - adding 49
  - brightness, adding 42
  - controlling presentation 44
  - cutting 49
  - decimals, adding 42
  - definition 26
  - deleting 50
  - determining data use 44
  - displaying partial 46
  - editing 49
  - emphasizing 46
  - expanding definitions 47
  - fill characters, using 45
  - filling fields with nulls 43
  - foreground color, adding 46
  - formatting 45
  - indicating scrolling direction 45
  - justifying data 42
  - length, specifying 42
  - limits 26
  - naming 41
  - pasting 49
  - position, specifying 41
  - prompting for selection values 44
  - protecting 43
  - reference tables, using 44
  - requiring values 43
  - reversing video 46
  - Selection Manager menu 93
  - showing content 43
  - skipping 43
  - storing data 41
  - supplying prompting 44
- Screen Help key 54
- screen help, adding 54
- screen identification segment 10
- Screen Manager menu 66
  - adding
    - functionality 83
    - menu items 82
  - creating 77
  - defining physical layout 79, 81
  - description of menu items 82
  - overview 78
  - predefined
    - screen requirements 78
    - screens 81
  - predefining screens 80
  - screens containing TITLE and VALUE 81
- screen requirements, Screen Manager menu 80
- Screen Table Painter
  - illustrated 28
  - invoking 27
  - screen layout 28
  - using 28
- screen tables 26
  - defined 2
  - defining 30
  - deleting 50
  - editing 49
  - excluding 14
  - information 11
  - maximum number of row occurrences 12
  - names 11
  - specifying 11
  - types 26
  - width, maximum 12
- screens
  - and menus 67
  - containing TITLE and VALUE 81
  - defined 2
  - layout
    - Screen Definer 8
    - Screen Table Painter 28
  - menus in 67
  - requirements, Selection Manager menu 91
  - verifying identification 10
- SCRIPT
  - commands in
    - field help screen 56
    - help for menus 112
    - screen help screen 54
  - tool 54, 56
- scripting help text 54, 56, 56

- scroll fields
  - field names [16, 17](#)
  - table names [16, 17](#)
- scrolling
  - in screen tables [12](#)
  - indicating direction [45](#)
- search libraries
  - Menu Manager menu [74](#)
  - Screen Manager menu [83](#)
  - Selection Manager menu [94](#)
  - Standard Session Manager menu [103](#)
- selecting global fields [38](#)
- Selection Manager menu
  - adding
    - functionality [94](#)
    - menu items [93](#)
  - creating [87](#)
  - defining physical layout [90, 92](#)
  - overview [88](#)
  - predefined
    - screen requirements [88](#)
    - screens [92](#)
  - predefining screens [91](#)
  - screen field names [93](#)
  - styles [66](#)
  - tables containing menu fields [92](#)
- setting, literal text display attributes [32](#)
- showing content of fields [43](#)
- side, specifying for user entry field [73](#)
- skipping fields [43](#)
- Specify Function Keys key [18](#)
- specifying
  - function key attributes [18](#)
  - position [41](#)
  - screen fields
    - length [42](#)
    - position [41, 41](#)
  - screen tables [11](#)
  - side for user entry field, Menu Manager menu [73](#)
  - titles, Menu Manager menu [72](#)
- Standard Session Manager menu [66](#)
  - adding
    - functionality [103](#)

- menu items [102](#)
- creating [97](#)
- defining
  - command area [102](#)
  - head area [101](#)
  - menu area [101](#)
  - session menu area [100](#)
- defining physical layout [99](#)
- description of menu items [102](#)
- modifying
  - names of menus [100](#)
  - screen table names [101, 101, 102](#)
- overview [98](#)
- starting new transactions
  - Menu Manager menu [74](#)
  - Screen Manager menu [83](#)
  - Selection Manager menu [94](#)
  - Standard Session Manager menu [103](#)
- statements
  - DISPLAY [116](#)
  - DISPLAY & TRANSFERCALL [116](#)
  - UNTIL...DISPLAY [116](#)
- storing
  - data [41](#)
- supplying prompting [44](#)
- support, contacting [xviii](#)

## T

- Table Definer, accessing [8](#)
- table fields
  - appending [37](#)
  - copying [36](#)
  - definition [34](#)
  - using definitions [36](#)
- tables
  - @COLOURS [60](#)
  - @SCREENCOLOURS [61](#)
  - @TMP\_FCNKEYS [18](#)
  - containing menu fields, Selection Manager menu [92](#)
  - names for scroll fields [16, 17](#)
  - screen [26](#)
- technical support [xviii](#)

- test flag
  - and menus 68
  - excluding 101
  - including 101
- Test key 109
- TEST primary command 109
- TEXTSETUP tool 59
- TIBCO\_HOME xv
- time field, including in menus 68
- time, displaying current
  - Menu Manager menu 73
  - Screen Manager menu 82
  - Selection Manager menu 93
  - Standard Session Manager menu 101, 102, 102
- title lines
  - bottom 72
  - highlighting 72
  - justifying 72
  - position 72
  - second 101
  - text 72
  - top 72
- title rows in screen tables 13
- tools
  - DEFINE\_MENU 72, 81, 92, 100
  - DEFINE\_OBJLIST 75, 84, 95, 104
  - DELETE\_DEFN 21, 111
  - DISPLAY\_MENU 117
  - for deleting definitions 21, 111
  - PROCESS\_FCNKEY 118
  - TEXTSETUP 56
- top title lines 72, 100
- types of prompts
  - Menu Manager menu 74, 75
  - Standard Session Manager menu 103

## U

- underline fields 46
- Unit field 10
- UNTIL...DISPLAY statement 116

- update mode
  - Menu Manager menu 74
  - Screen Manager menu 83
  - Selection Manager menu 94
  - Standard Session Manager menu 103
- user entry field, specifying side, Menu Manager menu 73
- user IDs
  - displaying
    - Menu Manager menu 72
    - Selection Manager menu 92
  - of creators 100
- users, prompting 44
- using
  - default fields 40
  - fill characters 45
  - global fields 38
  - predefined fields 36
  - PROCESS\_FCNKEY tool 118
  - reference tables 44
  - Screen Table Painter 28
  - table field definitions 36
  - virtual fields 63

## V

- VALIDATEFAIL exception 4
- validation
  - failure 4
  - Phase One 4
  - Phase Two 4
- Validation Exit key 16
- validation rules 13
- verifying screen identification 10
- vertical scrolling in screen tables 12
- virtual fields
  - \$MODIFIED 63
  - \$OCCNUM 64