

# **TIBCO® Object Service Broker**

## **National Language Support**

*Software Release 6.0  
July 2012*

## Important Information

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

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

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

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

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

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

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

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

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

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

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

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

TIBCO Software Inc. Confidential Information

# Contents

<b>Preface</b> .....	<b>vii</b>
Related Documentation .....	viii
TIBCO Object Service Broker Documentation .....	viii
Typographical Conventions .....	xiii
Connecting with TIBCO Resources .....	xvi
How to Join TIBCOCommunity .....	xvi
How to Access All TIBCO Documentation .....	xvi
How to Contact TIBCO Support .....	xvi
 <b>Chapter 1 National Language Support in the TIBCO Object Service Broker Environment</b> ....	<b>1</b>
Introduction .....	2
Advantages of NLS .....	2
Scope of NLS Support .....	2
NLS Functionality .....	4
Locale and Code Page Definitions .....	4
Locale Specifications .....	4
Folding Rules .....	5
Restrictions and Unsupported Items .....	6
 <b>Chapter 2 Character and Code Page Support</b> .....	<b>7</b>
Character Support .....	8
Character Sets .....	8
Code Points .....	8
Supported Code Pages .....	10
EBCDIC Code Pages .....	10
ASCII Code Page .....	10
TDS Code Pages .....	11
National Language Support for External Database Servers .....	11
 <b>Chapter 3 Data Interpretation</b> .....	<b>15</b>
Data Translation .....	16
Types of Data Translated .....	16
Translation of Data in Peer-to-Peer Communication .....	16
Translation for Remote Access .....	17
Mapping of Shaded Entries .....	17

Collation .....	19
Collating Sequence .....	19
Collation Behavior .....	19
Characters Affected .....	19
<b>Chapter 4 Encoded File Transfers .....</b>	<b>21</b>
Transferring Encoded Data Between Platforms .....	22
Requirements .....	22
The File Transfer Process .....	23
Step 1: Define the Input File .....	23
Step 2: Run S6BBRNLS/hrnbrnls Against the Input File .....	23
Step 3: Check Audit Log .....	24
Step 4: Run Sort on Output File, if Required .....	24
Step 5: Transfer Output File to Appropriate Platform .....	24
Step 6: Load Output File Using S6BBRTBL/hrnbrtbl Utility .....	24
Considerations .....	25
Constraints .....	25
<b>Chapter 5 Implementation Requirements .....</b>	<b>27</b>
Setup and Configuration .....	28
Responsibilities of the Database Administrator .....	28
Keyboard Support .....	28
Display Font Support .....	29
Changing the EBCDIC Display Code Page .....	31
Use of \$SETOPT .....	31
When to Use \$SETOPT .....	31
Special Considerations .....	32
Special Handling of 8-bit Bytes for Solaris .....	32
Restrictions .....	33
General Restrictions .....	33
Subview and Calculation Tables .....	33
Collation .....	33
Table Definitions .....	33
Naming Restrictions .....	34
NLS Items Not Supported .....	34
Client Restrictions .....	34
<b>Chapter 6 Specification of NLS Characteristics – @NLS1 Table .....</b>	<b>37</b>
@NLS1 Table Specification .....	38
Format of the @NLS1 Table .....	38
Activating the New Values .....	39
Enabling NLS .....	40

Additional locale.codepage Requirements . . . . .	40
Example Values for @NLS1 . . . . .	40
Restrictions on locale.codepage . . . . .	42
Values for SELF . . . . .	42
Values for REMOTE . . . . .	42
Values for ASCII . . . . .	43
Valid locale.codepage Specifications . . . . .	44
Example Specifications for locale.codepage . . . . .	46
Factors to Use for Determining Specifications . . . . .	46
Example 1. . . . .	46
Example 2. . . . .	47
Example 3. . . . .	47
<b>Chapter 7 Use of TDS Tables With a National EBCDIC Code Page . . . . .</b>	<b>49</b>
Use of the NLS bit . . . . .	50
Implementation . . . . .	50
Tables with NLS Bit Pre-set . . . . .	50
Setting the NLS Bit . . . . .	50
Expanded EBCDIC Code Page Substitution Characters . . . . .	52
Special Characters for TIBCO Object Service Broker Names . . . . .	52
Evaluation of TCP (TDS Code Page) versus SCP (System Code Page) Value . . . . .	53
Post-installation Peer NLS Testing . . . . .	54
Verification Test. . . . .	54
Windows Behavior . . . . .	54
<b>Chapter 8 Configuration for the Solaris Platform . . . . .</b>	<b>55</b>
Setup for the Solaris Platform . . . . .	56
Enabling Operation . . . . .	56
X-Windows Support and Keyboard Support . . . . .	57
X-Client Support . . . . .	57
Solaris Locale Support. . . . .	58
Customization. . . . .	58
Testing Your NLS Environment . . . . .	59
Verification . . . . .	59
Testing your NLS-enabled TIBCO Object Service Broker on Solaris . . . . .	59
<b>Appendix A NLS Supported Code Pages . . . . .</b>	<b>61</b>
EBCDIC Code Pages. . . . .	62
Code Page IBM-037 (ENGL, CDNB) . . . . .	63
Code Page IBM-273 (DEUT) . . . . .	63
Code Page IBM-277 (DANS, NORSE) . . . . .	64
Code Page IBM-278 (SUOM, SVEN) . . . . .	64

Code Page IBM-280 (ITAL) ..... 65

Code Page IBM-282 (PORT) ..... 65

Code Page IBM-284 (ESPA) ..... 66

Code Page IBM-285 (ENGB) ..... 66

Code Page IBM-297 (FRAN) ..... 67

Code Page IBM-500 (SCHW) ..... 67

ASCII Code Pages ..... 68

Code Page ISO8859-1 ..... 69

**Index ..... 71**

# 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 detailed information about how to implement National Language Support (NLS) in a TIBCO Object Service Broker environment.

## Topics

---

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

## Related Documentation

---

This section lists documentation resources you may find useful.

### TIBCO Object Service Broker Documentation

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

#### Fundamental Information

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

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



## Application Development and Management

The following manuals provide information about application development and management:

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

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

## System Administration on the z/OS Platform

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

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

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

## System Administration on Open Systems

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

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

## External Database Gateways

The following manuals describe external database gateways:

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

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

## Typographical Conventions

The following typographical conventions are used in this manual.

Table 1 General Typographical Conventions

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

Table 2 Syntax Typographical Conventions

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

Table 2 *Syntax Typographical Conventions*

Convention	Use
{ }	<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

# National Language Support in the TIBCO Object Service Broker Environment

This chapter describes the National Language Support (NLS) features and functionalities in TIBCO Object Service Broker.

## Topics

---

- [Introduction, page 2](#)
- [NLS Functionality, page 4](#)

## Introduction

---

This manual examines the National Language Support (NLS) features in TIBCO Object Service Broker. NLS provides users with the ability to enter, store, process, retrieve, display, and print character data in the language of their choice. You can use the characters present in Latin-1 (Western European) single-byte languages in:

- Data fields of TDS tables
- Input/output to display devices
- Data requests and transfers between TIBCO Object Service Broker peer servers

## Advantages of NLS

NLS provides:

- Flexibility

You can choose the code pages used for display devices, for the character data fields in TDS, and for the character data used in peer-to-peer communication.

- Enterprise-wide consistency

NLS ensures coherency in peer-to-peer communication by employing consistent rules for the mapping of lowercase characters to uppercase characters. In this manual, such mappings are referred to as folding.

- Data consistency

You can enter data from a display device using one National EBCDIC representation and store that data in a TIBCO Object Service Broker Data Object Broker in another National EBCDIC representation. This ensures data integrity in a distributed multi-national (European or Latin-1) environment.

## Scope of NLS Support

The following items are supported:

- Full access support for NLS-enhanced TDS tables: editing, browsing, searching, occurrence selection, sorting, and so on.
- Accented characters (for example, é and ü), where such items are used as strings and not as object names. They are supported in applications and in titles, field labels, menu bars, and button labels.

- Accented characters in generated reports, import files, and export files. However, in a TIBCO Object Service Broker z/OS environment using National EBCDIC, if system field and object names contain the special characters at sign (@), number sign (#), and dollar sign (\$), substitution characters could be produced for those characters.

# NLS Functionality

---

## Locale and Code Page Definitions

NLS functionality in the TIBCO Object Service Broker environment is dependent on the definition of a locale and a code page.

The locale and code page together function as an ordered pair of identifiers (*locale.codepage*). They represent the cultural conventions and code page properties used within the locale Data Object Broker (referred to as SELF) and when communicating with one or more peer Data Object Brokers (referred to as REMOTE).

### Locale

The locale is the portion of a user’s environment that is sensitive to cultural conventions of a particular language, country, or territory. The value defined for the locale establishes the language used for data manipulation and storage within a Data Object Broker.

### Code Page

The code page value identifies the character set encoding to be used. The notion of the code page value is used in several contexts:

Communication code page	This is associated with internode peer-to-peer user character data requests.
TDS code page	This is associated with the TIBCO Object Service Broker TDS character data fields.
System code page	This is associated with the TIBCO Object Service Broker Execution Environment (for example, code page IBM-037) and is used to encode object and field names in the metadata.

## Locale Specifications

In general, the specification of a locale determines such things as:

- The character set
- An appropriate message catalog

- National/regional conventions regarding upper/lowercasing
- Collation
- Date format
- Time format
- Monetary symbols and formats

## Folding Rules

For NLS support, you set a *locale.codepage* value. The locale setting currently affects only folding operations. For each valid locale value, the set of lowercase characters that are folded is defined in the following table:

Locale	Lowercase Characters Mapped to Uppercase
ENGL	a-z
ENGB	a-z
CDNB	a-z â à ç é ê ë è î ï ô û ü ù
DANS, NORS	a-z å ø æ
DEUT	a-z ä ö ü
ESPA	a-z ñ
FRAN	a-z
ITAL	a-z
PORT	a-z ã ç õ
SCHW	a-z ä ö ü
SUOM, SVEN	a-z ä å é ö

For example, “é”, listed in the table for the CDNB locale, is folded to “É”, and “í”, not listed in the table, results in “I”.

## Restrictions and Unsupported Items

There are some restrictions, as well as some NLS items that are not supported. Refer to [General Restrictions on page 33](#) and [NLS Items Not Supported on page 34](#) for more information.

## Chapter 2

# Character and Code Page Support

This chapter describes the characters and the code pages supported in TIBCO Object Service Broker.

## Topics

---

- [Character Support, page 8](#)
- [Supported Code Pages, page 10](#)

# Character Support

## Character Sets

The characters supported belong to the ISO8859-1 extended ASCII character set. Alternatively, this character set can be described as those characters belonging to the EBCDIC Country Extended Code Pages (CECP), for example, IBM-037 (U.S. EBCDIC). In either case, the description applies only to Latin-1 characters (Western European languages).

TIBCO Object Service Broker uses the following table as its default TDS code page.

### Code Page IBM-037 (U.S. English EBCDIC)

	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0					sp	&	-	ø	Ø	°	μ	^	{	}	\	0
-1					rsp	é	/	É	a	j	~	£	A	J	+	1
-2					â	ê	Â	Ê	b	k	s	¥	B	K	S	2
-3					ä	ë	Ä	Ë	c	l	t	·	C	L	T	3
-4					à	è	À	È	d	m	u	©	D	M	U	4
-5					á	í	Á	Í	e	n	v	§	E	N	V	5
-6					â	î	Â	Î	f	o	w	¶	F	O	W	6
-7					ã	ï	Ã	Ï	g	p	x	¼	G	P	X	7
-8					ç	ì	Ç	Ì	h	q	y	½	H	Q	Y	8
-9					ñ	ß	Ñ	·	ı	r	z	¾	I	R	Z	9
-A					ø	ı	ı	:	«	ª	j	[	-	¹	²	³
-B					.	\$	,	#	»	º	ı	]	ó	ú	ô	û
-C					<	*	%	@	ð	æ	Đ	—	ó	ú	ô	û
-D					(	)	_	'	ý	,	Ý	"	ó	ú	ô	û
-E					+	;	>	=	þ	Æ	Ɔ	'	ó	ú	ô	û
-F						¬	?	"	±	α	®	x	ó	ý	ô	

## Code Points

The encoding that represents a character on a page is known as the code point. Note the following regarding code points:

- To determine the code point of a character, use the column heading as the high-order nibble of the byte, and the row heading as the low-order nibble of the byte. For example, the code point for the plus/minus character (±) is 0x8F.



- Code points 0x00 to 0x3F and 0xFF are for non-printing characters, that is, control characters.
- Code point 0x40 is the space character (sp). Code point 0x41 is the required space character (rsp).
- Code point 0x60 is the hyphen character (-). Code point 0xCA is the non-breaking hyphen character (-).
- The shaded table entries represent control characters or invalid code points.

# Supported Code Pages

## EBCDIC Code Pages

The following EBCDIC (CECP) code pages are supported:

Code Pages Without Euro	Code Pages With Euro	Character Set
IBM-037	IBM-1140	ENGL, CDNB
IBM-273	IBM-1141	DEUT
IBM-277	IBM-1142	DANS, NORS
IBM-278	IBM-1143	SUOM, SVEN
IBM-280	IBM-1144	ITAL
IBM-282		PORT
IBM-284	IBM-1145	ESPA
IBM-285	IBM-1146	ENGB
IBM-297	IBM-1147	FRAN
IBM-500	IBM-1148	SCHW

## ASCII Code Page

The following ASCII code pages are supported: ISO8859-1, ISO8859-15, and WIN-1252. These code pages are supported for ASCII display and communication and are used by Windows and Solaris.

- These code pages remap the ASCII to EBCDIC translation of two characters. The two characters are the EBCDIC logical-not character (¬), which is remapped to the ASCII circumflex (^), and the EBCDIC pipe character (|), which is remapped to the ASCII bar character (|). With the exception of these special characters, any character can be entered in user data fields.
- The ISO8859-1 and ISO8859-15 code pages, as used in TIBCO Object Service Broker, are subsets of the Windows CP1252 (default) character set. Refer to [Code Page ISO8859-1 on page 69](#) for a copy of the ISO8859-1 code page.

- All ISO-8859-1 characters use the same code points as those in CP1252. However, CP1252 contains characters that do not exist in ISO-8859-1 (or in Latin-1 EBCDIC). When TIBCO Object Service Broker encounters an unsupported (ASCII) character, it performs a reciprocal translation; that is, provided the input ASCII and output ASCII are both CP1252, the desired character is preserved. For example, the euro symbol, which does not exist in either IBM-037 or ISO-8859-1 and which is at x'80' in CP1252, is translated to a unique non-display code point in IBM-037 EBCDIC and back to x'80' on output to CP1252.  
  
The WIN-1252 code page is recommended since it contains code points for both the Universal Currency Symbol (UCS) at x'A4' and the euro at x'80', while the ISO-8859-1 code page and the ISO-8859-15 code page are missing one or the other.

TDS Code Pages

The valid TDS code pages are as follows:

Platform	Code Page
Windows and Solaris	IBM-037 or IBM-1140
z/OS	Any supported EBCDIC code page

National Language Support for External Database Servers

You can connect an external database server on one platform to a Data Object Broker running on the other platform using a different code page.

To specify the default code pages for specific table types or server IDs, use the @SERVERCONFIG table. In this parameterized table:

- The “\_NT” instance identifies a code page for a specific server type, for example, DB2
- The “\_NS” instance identifies a code page for a specific server ID, for example DB2PROD

Adding a Server Type

To add a server type to the table:

1. Type @SERVERCONFIG at the ED edit table option on the workbench and press Enter.

The following appears:

EDITING TABLE : @SERVERCONFIG

```
ENTER  PARM  VALUE  SERVERTYPE      :
ENTER  PARM  VALUE  LOCATION        :
```

2. Type\_NT for the SERVERTYPE value.

The following appears:

```

EDITING TABLE      :    @SERVERCONFIG(_NT)
COMMAND ==>

```

SCROLL: P

NAME	TYPE	SYNTAX	LENGTH	DECIMAL
-----	-	-	-----	-----

3. To get a new line for data entry, type an i in the command field to the left of the NAME column.
4. Type the information for your server type, pressing PF11 as required to see the other fields available.

The information should be the same as the example below, replacing NAME and VALUE with the appropriate information for your server type.

```

--- SINGLE OCCURRENCE EDITOR ---
EDITING TABLE   : @SERVERCONFIG
TABLE TYPE      : TDS
COMMAND ==>

```

```
NAME          : DB2
TYPE          :
SYNTAX        : C
LENGTH        :      32
DECIMAL       :      0
VALUE         : ESPA.IBM-284
MODIFY        : N
ALLOWED_VALUES :
```

## Adding a Server ID

To enter information for a server ID, use the same procedure with “\_NS” as the server type option.

The information should be the same as the example below, replacing NAME and VALUE with the appropriate information for your server ID.

---

```

                                --- SINGLE OCCURRENCE EDITOR ---
EDITING TABLE      : @SERVERCONFIG
TABLE TYPE         : TDS
COMMAND ==>

```

```

-----
NAME                : DB2PROD
TYPE                :
SYNTAX              : C
LENGTH              :      32
DECIMAL             :      0
VALUE               : ESPA.IBM-284
MODIFY              : N
ALLOWED_VALUES      :

```

---



## Chapter 3      **Data Interpretation**

This chapter describes the types of data values translated and the collating sequence in TIBCO Object Service Broker.

### Topics

---

- [Data Translation, page 16](#)
- [Collation, page 19](#)

## Data Translation

---

### Types of Data Translated

If data translations are being performed on peer-to-peer (remote) communication, the following types of data values are translated:

- Data parameters
- Data values (either literals or data values substituted from an occurrence) in selection strings
- Data values in the occurrences provided with an INSERT, DELETE, or REPLACE request
- Data values in a retrieved occurrence



The NLS shareable tool enables a database administrator to set code page values in translation tables. See *TIBCO Object Service Broker Shareable Tools* for information about using the NLS tool.

### Translation of Non-character Data

Non-character values created by tools, utilities, or application code that are stored in character string variables are treated as characters and are subject to translation from one code page to another as required. The results of such translation are defined when the data is considered as characters and are otherwise undefined. In particular, the string results produced by the [GENBIN](#) scalar shareable tool are undefined if they undergo code page translation.

### Translation of Data in Peer-to-Peer Communication

Translations performed on customer data for peer-to-peer communication are faithful. The code point used to represent a character on the first Data Object Broker is translated to the corresponding code point to represent the same character on the second Data Object Broker, if both of the following occur:

- A Data Object Broker transmits data to a second Data Object Broker that uses a different code page.
- The communication code page is set correctly.



## Example

Data Object Broker1 uses IBM-037 (U.S. English) and Data Object Broker2 uses IBM-280 (Italian). If a string of customer data contains the “a with grave accent” character (à), the representation on Data Object Broker1 (0x44) is translated to 0xC0 when the data is transmitted to Data Object Broker2.

Maintaining a faithful translation ensures that the operation is reversible. Data translated from code page CP1 (IBM-037) to code page CP2 (IBM-280) and then back again appears exactly as it did originally.

## Translation for Remote Access

For a remote request/response, a translation is performed on a data value if all the following conditions are satisfied:

- The TCP value is not the same as the communication code page value.
- The request is a data access request (GET, FORALL, INSERT, DELETE, or REPLACE) that is not for a CTABLE.
- The table’s NLS bit is not set (that is, the table uses a TDS code page as opposed to a system code page).
- The semantic data type is identifier, string, or untyped.
- The syntax type is fixed-length character (C) or variable-length character (V).

## Mapping of Shaded Entries

The mapping of shaded table entries (control and illegal characters) from one code page to another is implementation dependent and is subject to change.<sup>1</sup> However, the translation of a code point from one Latin-1 EBCDIC code page to another Latin-1 EBCDIC code page results in the same encoded value. For example, translating 0xFF from IBM-037 EBCDIC to IBM-500 EBCDIC results in the value 0xFF.

## Binary Values

The specification of binary values corresponding to shaded table entries in character fields is also implementation dependent. The translation of such a binary value from one Latin-1 EBCDIC code page to another Latin-1 EBCDIC code page results in the same encoded value. However, the mapping between a

---

1. Refer to the code page IBM-037, illustrated in [Character Support on page 8](#), for an example of shaded entries.

Latin-1 EBCDIC code page and the ISO8859-1 code page, or the reverse mapping from the ISO8859-1 code page to the Latin-1 EBCDIC code page is undefined. This could, for example, invalidate ASCII printer control strings (for example, LaserJet PCL sequences) stored in TIBCO Object Service Broker.

# Collation

---

## Collating Sequence

The collating sequence of a remote peer request can differ from that of a local request for the same data if the code page of the remote TDS table differs from that of the local TDS table.

### Example

For example, consider the French words *école* and *femme*. When represented in code page IBM-037, the first letter of each word is represented as code points 0x51 and 0x86. However, in the French EBCDIC code page IBM-297 the first letters are represented as code points 0xC0 and 0x86.

## Collation Behavior

For sorting character data, TIBCO Object Service Broker uses the binary machine representation of the data items to determine the sort order. Therefore, the following inconsistent sorting behavior occurs:

- Using the IBM-037 code page, *école* is sorted before *femme* ('*école*' < '*femme*')
- Using the IBM-297 code page, *femme* is sorted before *école* ('*femme*' < '*école*')

This collation behavior is an important consideration in requests for character data in explicit or implicit sort operations and can impact peer requests across systems with differing values of the TDS code page.

## Characters Affected

In general, the collating behavior of characters in the English alphabet (a-z, A-Z) and the numeric characters 0-9 are the same in all currently supported EBCDIC code pages. Characters other than these could collate differently when represented in different EBCDIC code pages.



## Chapter 4      **Encoded File Transfers**

This chapter describes how table data can be moved between two TIBCO Object Service Broker Pagestores.

### Topics

---

- [Transferring Encoded Data Between Platforms, page 22](#)
- [The File Transfer Process, page 23](#)
- [Considerations, page 25](#)

## Transferring Encoded Data Between Platforms

---

Table data can be moved between two TIBCO Object Service Broker Pagestores, that is, z/OS and Windows or Solaris platforms, whose data are encoded using different code pages. This can be done with the help of the S6BBRNLS/hrnbrnls utility, which changes the character fields in row occurrences from one code page representation to another.

The following sections describe the requirements and procedure and discuss various restrictions and considerations.

### Requirements

Note the following:

- Input files must be unloaded from a TIBCO Object Service Broker table using one of the offline batch unload utilities (S6BBRULB/hrnbrulb, S6BBRULH/hrnbrulh, or S6BBRULA/hrnbrula).
- The data in the unload file must be in EBCDIC format and must be in the correct sequence (ascending—parameters 1 through *n*; primary keys 1 through *n*).
- If translation is being done on Windows or Solaris, the translation table must be on that platform.

**See Also** *TIBCO Object Service Broker for z/OS Utilities* or *TIBCO Object Service Broker for Open Systems Utilities* for information about using the S6BBRNLS/hrnbrnls, S6BBRULB/hrnbrulb, S6BBRULH/hrnbrulh, and S6BBRULA/hrnbrula utilities.

## The File Transfer Process

---

The file transfer process involves using a translation utility, S6BBRNLS/hrnbrnls, that changes the character fields in row occurrences from one code page representation to another. The procedure is as follows:

1. Define the input file.
2. Run the S6BBRNLS/hrnbrnls utility against the input file.
3. Check the audit log produced.
4. Run sort on the output file, if required.
5. Transfer the output file to the appropriate platform.
6. Load the file using the S6BBRTBL/hrnbrtbl utility.

Further details on each of these steps are provided in the following sections.

### Step 1: Define the Input File

Define the input file by using the [BATCHLOAD\\_CARDS](#) tool to create a control file. For more information on this tool, refer to *TIBCO Object Service Broker Shareable Tools*. The control file can also be created manually using a text editor.

When building the control file, there are several considerations that should be taken into account. For more information, refer to the S6BBRNLS/hrnbrnls utility as described in *TIBCO Object Service Broker for z/OS Utilities* or *TIBCO Object Service Broker for Open Systems Utilities*.

### Step 2: Run S6BBRNLS/hrnbrnls Against the Input File

Using the control file, run S6BBRNLS/hrnbrnls against the input file. For detailed information on the S6BBRNLS/hrnbrnls utility, refer to *TIBCO Object Service Broker for z/OS Utilities* or *TIBCO Object Service Broker for Open Systems Utilities*.



The code page specified for the input file is checked to ensure it is supported by TIBCO Object Service Broker. If not, the request is rejected.

### Step 3: Check Audit Log

As part of the process, an audit log file is produced. This file is an activity report containing information and warning and error messages as well as run statistics. Always inspect this report before using the output file in another process. Should you encounter problems requiring TIBCO Support services, the audit log is always be required (either hardcopy or file).

### Step 4: Run Sort on Output File, if Required

If S6BBRNLS/hrnbrnls indicates that translation changed the sequence of the translated file, a sort is necessary before the file can be loaded using S6BBRTBL/hrnbrtbl. If no sort is available on Windows or Solaris, you must translate the file on the z/OS platform and sort it prior to using File Transfer Protocol (FTP) to move the file back to Windows or Solaris.



If a field is eligible for translation and is also a parameter or primary key field, be aware that the output file could require sorting after translation. If such fields exist in the input file, we recommended that translation and sorting be done on z/OS prior to porting to Windows or Solaris if the file is to be ported in this way.

### Step 5: Transfer Output File to Appropriate Platform

Copy the output files to the destination platform or move the file using File Transfer Protocol (FTP).

### Step 6: Load Output File Using S6BBRTBL/hrnbrtbl Utility

For information on the S6BBRTBL/hrnbrnls utility and its use, refer to *TIBCO Object Service Broker for z/OS Utilities* or *TIBCO Object Service Broker for Open Systems Utilities*.



## Considerations

---

### Constraints

Note the following:

- The S6BBRNLS/hrnbrnls utility does not perform folding on character data being translated.
- Do not translate system table data; it is encoded using the system code page (SCP). Attempts to translate system tables result in an error message in the audit log.

For more detailed information about possible constraints and other considerations, refer to the S6BBRNLS/hrnbrnls utility as described in *TIBCO Object Service Broker for z/OS Utilities* or *TIBCO Object Service Broker for Open Systems Utilities*.



Do not use the S6BBRNLS/hrnbrnls utility to convert a file from ASCII to EBCDIC, as there could be problems with numeric fields. Potentially, the resulting file could be unusable.



## Chapter 5      **Implementation Requirements**

This chapter describes how to configure the TIBCO Object Service Broker environment, the keyboard, and the end user display for the use of NLS.

### Topics

---

- [Setup and Configuration, page 28](#)
- [Changing the EBCDIC Display Code Page, page 31](#)
- [Special Considerations, page 32](#)
- [Restrictions, page 33](#)

## Setup and Configuration

---

The following information describes the requirements for configuring the TIBCO Object Service Broker environment, the keyboard, and the end user display for the use of NLS.

### Responsibilities of the Database Administrator

The database administrator is responsible for ensuring that an enterprise is properly enabled for NLS. This involves the following tasks:

- Decide which specifications of a TDS code page and a communication code page are appropriate for a peer-connected enterprise. Refer to [Restrictions on locale.codepage on page 42](#) for the list of valid specifications.
- Ensure all nodes in a peer-connected enterprise have similar values for the TDS code page and the communication code page.
- Ensure all peers have suitably initialized entries in their local instances of the @NLS1 table to correctly enable NLS support. For more information on the @NLS1 table, refer to [Chapter 6, Specification of NLS Characteristics – @NLS1 Table, on page 37](#).
- After configuring @NLS1 for each peer, log in to each node as a normal user to validate that NLS initialization completes normally and without errors.
- Set PATH as described in [Data Translation on page 16](#).

### Keyboard Support

NLS supports the full set of Latin-1 keyboard characters as described previously. The ability to exploit the keyboard is dependent upon the host operating system's ability to recognize these characters. The end user must perform the necessary system configuration or setup to support the national keyboard. Guidelines for each host operating system follow.



Failure to properly configure the (national) keyboard can result in unexpected program behavior.

### Windows

To set up national keyboard support, proceed as follows from the Start Menu:

1. Select the Settings icon and then select the Control Panel icon.
2. Select the Keyboard icon and select the Input Locales tab.

3. Select the input locale whose keyboard layout you want to change and then select Properties.
4. Choose the national keyboard that most closely matches the one attached to your system (for example, U.S. English, French, or Swedish).

This could require that the original system setup disks or CD-ROM be available to install additional keyboard drivers.

Under Windows, you can reconfigure your keyboard driver dynamically without exiting a session. In this way you can enter multi-national characters from a single keyboard session.

## Solaris

Refer to your Solaris documentation regarding exploitation of national keyboards.

## z/OS

By default, z/OS display devices attached to conforming display controllers provide identification that defines their supported display code page. NLS uses this code page to configure a suitable mapping between the display code page and the TDS code page. If you are using a 3270 emulator, consult your documentation to determine whether this identification is provided under software control.

If your display device does not provide this support, the TIBCO Object Service Broker initialization assumes that the display code page is the same as the TDS code page. If this is not the case (for example, connecting Swedish EBCDIC devices to a German Data Object Broker), you must provide additional customization. Refer to [Changing the EBCDIC Display Code Page on page 31](#).

## Display Font Support

To display accented characters, the selected font must be able to display the ASCII code page, or equally, the full national EBCDIC code page:

## Windows

Change the text and graphical display client fonts to an ISO8859-1 font (for example, Courier, Courier New, Fixedsys, Lucida Console, or Modern). The installation default font is Terminal (a DOS font). In addition to choosing an ISO8859-1 font, consider choosing one that is also fixed-pitch and TrueType.



Failure to properly configure the display font under Windows can result in unexpected display results.

## **Solaris**

The default X-Windows display font is typically an ISO8859-1 font, so no action should be required on your part.

## **z/OS**

Refer to [Keyboard Support on page 28](#), specifically the discussion on keyboard support for Windows.

## Changing the EBCDIC Display Code Page

---

To configure a z/OS EBCDIC display device to use a code page other than the TDS value (that is, IBM-037), use the [\\$SETOPT](#) tool. This tool, which sets session parameters, has two arguments, *parameter* and *value*.

### Use of \$SETOPT

To use [\\$SETOPT](#), call the tool within a rule. For example, to use a French EBCDIC device attached to a German z/OS system, run a rule similar to the following:

```
CALL $SETOPT( 'DISPLAYCODEPAGE', 'FRAN.IBM-297' );
```

This calls an external translation table that maps French EBCDIC characters in code page IBM-297 into German EBCDIC in code page IBM-273 for use by the TIBCO Object Service Broker system. This translation ensures that data stored in the enterprise TDS is correct, and prevents data pollution from multiple devices in differing code pages.

The second argument to [\\$SETOPT](#) can alternately be specified in the form ". IBM-297" with a leading period.

### When to Use \$SETOPT

Use the [\\$SETOPT](#) tool only if NLS cannot determine the code page of the display device and if that code page differs from the one defined in the SELF entry of the @NLS1 table. For more information on the @NLS1 table, refer to [@NLS1 Table Specification on page 38](#).

[\\$SETOPT](#) is particularly useful for European multi-national customers with a central corporate Data Object Broker in one country and end users in other countries with differing displays. For example, it can be used by traveling users accustomed to a French EBCDIC display at a home location who need to use a Swedish EBCDIC display at a remote location.

See Also *TIBCO Object Service Broker Shareable Tools* for more information about the [\\$SETOPT](#) tool.

## Special Considerations

---

### Special Handling of 8-bit Bytes for Solaris

To fully support the expanded character set, Solaris systems must be enabled for 8-bit bytes rather than the traditional 7-bit bytes. System-specific commands and/or setup could be required. Consult your system administrator. As a start, consider the following suggestions:

- To allow 8-bit ASCII characters to be received from a terminal, enter the command:  
`stty -istrip -parity`
- To exploit national language message catalogs, and such, you must initialize the environment variable `LC_TYPE`.
- Ensure that you can enter accented characters and have them echoed from the Solaris command line before trying to run NLS-enabled TIBCO Object Service Broker. Consult your system administrator for additional assistance.

Refer to [Setup for the Solaris Platform on page 56](#) for more information.



# Restrictions

---

## General Restrictions

The following restrictions apply to NLS-enabled TIBCO Object Service Broker:

- NLS does not set the decimal separator. This is done through the DECIMALSEPARATOR session parameter.
- The [LOWERCASE](#) tool does not map accented characters to lowercase. It currently changes only letters A-Z to lowercase. This applies to all supported platforms.

## Subview and Calculation Tables

Support of peer data requests is provisional:

- Subview (SUB) and calculation (CLC) tables defined on TIBCO Object Service Broker system tables are not supported in peer data requests. Such requests could return invalid data.
- Subview (SUB) and calculation (CLC) tables defined on user-defined tables are supported in peer requests. This implies that character fields can undergo a code page translation.

## Collation

A set of characters defined on a given code page could sort (collate) differently when translated to a second code page. Refer to [Collation on page 19](#) for more information.

## Table Definitions

Table definitions that include default values for character fields could produce incorrect results if the table definition is transferred between Data Object Brokers with different TDS code page values.

## Naming Restrictions

To enable interoperability of NLS-enabled TIBCO Object Service Broker Data Object Brokers, the following naming restrictions apply:

- Standard TIBCO Object Service Broker node name conventions apply, that is, a maximum of 16 alphanumeric characters, case insensitive (that is, A-Z and 0-9).
- TIBCO Object Service Broker user IDs and passwords are restricted to the characters a-z, A-Z, and 0-9. Blank ( ' ') is supported on z/OS platforms. Use of other characters can result in undefined behavior in NLS-enabled environments.

## NLS Items Not Supported

The following exceptions must be considered when implementing NLS in TIBCO Object Service Broker:

- Syntax W data items are not supported on Windows or Solaris.
- Accented characters are not valid in TIBCO Object Service Broker object, field, and parameter names, except when used as substitution characters for the special characters at sign (@), number sign (#), and dollar sign (\$). Refer to [Expanded EBCDIC Code Page Substitution Characters on page 52](#) for a list of the supported substitution characters.
- The operation of the LOAD and UNLOAD tools (both the interactive and batch versions) is not defined when the TDS code page of the unloaded system's data does not match that of the system being loaded with the data. This restriction applies whether the TDS code page is implicit (non-NLS) or explicit (NLS-enabled).
- The use of accented or national characters is not supported in objects that are used to form system-dependent names (such as filenames).

## Client Restrictions

To enable interoperability of NLS-enabled TIBCO Object Service Broker Data Object Brokers, the following naming restrictions apply:

- Standard TIBCO Object Service Broker node name conventions apply, that is, a maximum of 16 alphanumeric characters, case insensitive (that is, A-Z and 0-9).
- TIBCO Object Service Broker user IDs and passwords are restricted to the characters a-z, A-Z, and 0-9. Blank ( ' ') is supported on z/OS platforms. Use

of other characters can result in undefined behavior in NLS-enabled environments.

**See Also**     *TIBCO Object Service Broker Shareable Tools* for information about the [LOAD](#), [UNLOAD](#), and [LOWERCASE](#) tools.

*TIBCO Object Service Broker for z/OS Utilities* or *TIBCO Object Service Broker for Open Systems Utilities* for information about the offline load and unload tools.

*TIBCO Object Service Broker Parameters* for information about the `DECIMALSEPARATOR` parameter.



## Chapter 6

## Specification of NLS Characteristics – @NLS1 Table

This chapter describes how to configure a TIBCO Object Service Broker node to use NLS.

### Topics

---

- [@NLS1 Table Specification, page 38](#)
- [Enabling NLS, page 40](#)
- [Restrictions on locale.codepage, page 42](#)
- [Example Specifications for locale.codepage, page 46](#)

## @NLS1 Table Specification

To allow for the NLS configuration of a TIBCO Object Service Broker node (Data Object Broker), the customer locale and code page information must be specified at system installation. These specifications appear in the @NLS1 TDS table. The two occurrences of this table (SELF and REMOTE) currently define:

- The *locale.codepage* for user character data stored in user TDS tables in the local Data Object Broker (SELF).
- The *locale.codepage* to be used for character data in peer-to-peer communication (REMOTE)

The database administrator (or another level-7 user) is responsible for correctly defining the contents of the @NLS1 table for each Data Object Broker in the enterprise.



An incorrect specification can result in unexpected behavior and/or data corruption on one or more Data Object Brokers.

### Format of the @NLS1 Table

The table @NLS1 is parameterized by location. Explanations of each field follow.

Key	A key value that can be any valid binary 4 integer.
CompType	A component type, the value of which is SELF, REMOTE, or ASCII.
CompName	Component name; value reserved for future use.
Locale_CP	The locale + code page; can be any valid <i>locale.codepage</i> value, for example, SVEN.IBM-278, ENGL.IBM-037.

## Activating the New Values

The @NLS1 table is bound in session storage. To unbind the table and reset the values, complete the following steps:

1. Obtain the unbound version of the @NLS1 table.
  - In a z/OS TSO environment, log out and log in again to a TIBCO Object Service Broker session to obtain the unbound version of the table.
  - In all other environments, shut down and restart the Execution Environment, and then restart the session to obtain the unbound version of the table.
2. Modify the @NLS1 table as required.
3. Optionally rebind the @NLS1 table.
  - In a z/OS TSO environment, log out and log in again to a TIBCO Object Service Broker session to obtain the revised NLS settings. For peer-to-peer connections, you must also shut down and restart the peer servers to obtain the revised NLS settings.
  - In all other environments, shut down and restart the Execution Environment, and then restart the session to obtain the revised NLS settings. Peer servers are re-initialized for you when the Execution Environment is restarted.

**See Also**     *TIBCO Object Service Broker Application Administration* for more information about binding and unbinding tables.

## Enabling NLS

---

When installed, the @NLS1 table is defined and contains no occurrences. With @NLS1 in this state, a TIBCO Object Service Broker system has NLS disabled. To enable NLS, the SELF and REMOTE occurrences must exist and contain valid entries. The ASCII occurrence is optional. It is ignored on z/OS. For Windows and Solaris, it must contain a valid entry if present.

- A valid entry for the SELF occurrence defines the *locale.codepage* for the data stored in the Data Object Broker; that is, the folding rules to be used and the explicit code page used to represent character data in user-defined TDS tables (the TDS code page).
- A valid entry for the REMOTE occurrence defines the *locale.codepage* for data transmitted in peer-to-peer communication (communication code page).
- A valid entry for the ASCII occurrence defines the *codepage* used for ASCII data.

### Additional locale.codepage Requirements

The following requirements and conventions apply:

- The *locale.codepage* specification must be left justified in the field with no leading blanks. Alphabetic letters must be uppercase.
- An explicit code page specification without a leading locale part is valid if a leading period precedes the code page. For example, the form .IBM-278 is valid and is considered equivalent to the specification ENGL.IBM-278.
- A specification of ENGL for the locale is valid for all valid code page parts, enabling such specifications as ENGL.IBM-278. This example defines the semantics for the ENGL locale (folding rules) applied to the IBM-278 Swedish EBCDIC code page.

### Example Values for @NLS1

#### Example 1

A suitable set of values for an English language installation of TIBCO Object Service Broker in the U.S. is:

```
1, 'SELF' , '' , 'ENGL.IBM-037'
2, 'REMOTE' , '' , 'ENGL.IBM-037'
```



**Example 2**

A suitable set of values for a Swedish installation of TIBCO Object Service Broker running on Windows is:

- 1, 'SELF' , ' ' , 'SVEN.IBM-037'
- 2, 'REMOTE' , ' ' , 'SVEN.IBM-037'

## Restrictions on locale.codepage

---

The values entered for the *locale.codepage* for SELF and REMOTE must (when default values are resolved as described) specify a locale part that is consistent with the code page specification.



If the value is not specified as described in [Enabling NLS on page 40](#), the login session fails NLS initialization and terminates the session. A database administrator (or other level-7 user) must modify entries in the @NLS1 table to create or restore a correct and valid state.

### Values for SELF

#### Code page

The code page part for SELF must correctly specify the current code page of the customer TDS character data. If you specify another value (for example, ENGL.IBM-037), some character data could be corrupted and character strings could be folded incorrectly.



For Open Systems, the only valid code page parts for the SELF occurrence are IBM-037 and IBM-1140. It is the responsibility of the enterprise database administrator to ensure this requirement is met. Failure to do so can result in data corruption or NLS initialization failure.

#### Locale

The locale part for SELF must be the same value as the locale part for REMOTE.

### Values for REMOTE

#### Code page

The code page part for the REMOTE occurrence of each peer Data Object Broker in an enterprise-wide cluster must (when default values are resolved as described) specify the same value.



The enterprise database administrator must ensure that this requirement is met. Failure to ensure equivalence can result in peer data corruption and/or the inability to interoperate (for example, an access error could occur on peer data requests).

## Locale

The locale part for REMOTE on each peer Data Object Broker in an enterprise-wide cluster must (when default values are resolved as described) specify the same value.



The enterprise database administrator must ensure that this requirement is met. Failure to ensure equivalence can result in peer data corruption and/or the inability to interoperate (for example, an access error could occur on peer data requests).

## Values for ASCII

### Code page

The code page part for the ASCII occurrence describes the ASCII code page to be used to display ASCII data. This code page must be compatible with the TDS code page specified in the SELF occurrence. For a TDS code page of IBM-037, either ISO-8859-1 or WIN-1252 is valid for the ASCII code page. For a TDS code page of IBM-1140, either ISO-8859-15 or WIN-1252 is valid for the ASCII code page.

Valid locale.codepage Specifications

The following table lists valid *locale.codepage* specifications:

locale.codepage	Language	Territory/Country
CDNB.IBM-037 or CDNB.IBM-1140	Canadian Bilingual	Canada
DANS.IBM-037 or DANS.IBM-1140	Danish	Denmark
DANS.IBM-277 or DANS.IBM-1142	Danish	Denmark
DEUT.IBM-037 or DEUT.IBM-1140	German	Austria, Germany
DEUT.IBM-273 or DEUT.IBM-1141	German	Austria, Germany
ENGB.IBM-037 or ENGB.IBM-1140	English	Great Britain
ENGB.IBM-285 or ENGB.IBM-1146	English	Great Britain
ENGL.IBM-037 or ENGL.IBM-1140	English	USA
ESPA.IBM-037 or ESPA.IBM-1140	Spanish	Latin America, Spain
ESPA.IBM-284 or ESPA.IBM-1145	Spanish	Latin America, Spain
FRAN.IBM-037 or FRAN.IBM-1140	French	France
FRAN.IBM-297 or FRAN.IBM-1147	French	France
ITAL.IBM-037 or ITAL.IBM-1140	Italian	Italy
ITAL.IBM-280 or ITAL.IBM-1144	Italian	Italy
NORS.IBM-037 or NORS.IBM-1140	Norwegian	Norway
NORS.IBM-277 or NORS.IBM-1142	Norwegian	Norway
PORT. IBM-037 or PORT.IBM-1140	Portuguese	Portugal
PORT. IBM-282	Portuguese	Portugal
SCHW.IBM-037 or SCHW.IBM-1140	French, German	Switzerland
SCHW.IBM-500 or SCHW.IBM-1148	French, German	Switzerland
SUOM.IBM-037 or SUOM.IBM-1140	Finnish	Finland

locale.codepage	Language	Territory/Country
SUOM.IBM-278 or SUOM.IBM-1143	Finnish	Finland
SVEN.IBM-037 or SVEN.IBM-1140	Swedish	Sweden
SVEN.IBM-278 or SVEN.IBM-1143	Swedish	Sweden

In addition, the ENGL locale is supported with all the above code pages (for example, ENGL.IBM-278).

## Example Specifications for locale.codepage

### Factors to Use for Determining Specifications

You should try to exploit any similarity of specifications to minimize the overhead of character data code page translation for remote accesses. The best choice depends on the following:

- Data distribution characteristics
- [Example 2](#) explains the consequences of not applying this consideration.
- Access request frequencies of the enterprise-wide data
- Relative performance of the hardware used to support each of the Data Object Brokers

### Example 1

The following is a valid specification for a cluster of three nodes running in Swedish:

Platform	Node Name	SELF	REMOTE	Translation Required
Windows	NODE1	SVEN.IBM-037	SVEN.IBM-278	Y
z/OS	NODE2	SVEN.IBM-278	SVEN.IBM-278	N
Solaris	NODE3	SVEN.IBM-037	SVEN.IBM-278	Y

In the above example, NODE1 and NODE3 perform translations when they make requests from or respond to either of the other Data Object Brokers; NODE2 performs no translations.

### Example 2

The following specification results in more processing, because it fails to exploit any commonality of code page representation. It forces each Data Object Broker to perform translations on each remote request and on each response to a remote request.

Platform	Node Name	SELF	REMOTE	Translation Required
Windows	NODE1	SVEN.IBM-037	SVEN.IBM-500	Y
z/OS	NODE2	SVEN.IBM-278	SVEN.IBM-500	Y
Solaris	NODE3	SVEN.IBM-037	SVEN.IBM-500	Y

Unless there is some compelling reason to use IBM-500 as the communication code page, it would be more efficient to choose some other enterprise-wide communication code page value.



The above specification is invalid as SVEN.IBM-500 is not a correct *locale.codepage*. It is used for illustration purposes only.

### Example 3

Using the same scenario, IBM-037 could be a better choice as a remote value. With this specification, only the z/OS node performs code page translations for remote requests/responses.

Platform	Node Name	SELF	REMOTE	Translation Required
Windows	NODE1	SVEN.IBM-037	SVEN.IBM-037	N
z/OS	NODE2	SVEN.IBM-278	SVEN.IBM-037	Y
Solaris	NODE3	SVEN.IBM-037	SVEN.IBM-037	N





## Chapter 7

# Use of TDS Tables With a National EBCDIC Code Page

This chapter describes how to use TDS tables with a national EBCDIC code page.

## Topics

---

- [Use of the NLS bit, page 50](#)
- [Expanded EBCDIC Code Page Substitution Characters, page 52](#)
- [Post-installation Peer NLS Testing, page 54](#)

## Use of the NLS bit

---

The TIBCO Object Service Broker metadata (the definitions, characteristics, access paths, and storage locations of all TIBCO Object Service Broker data and programs) is represented using the TIBCO Object Service Broker system code page. TIBCO Object Service Broker uses the NLS bit to recognize tables that are exempt from NLS translation, as are the tables of the metadata. The NLS bit is part of the RESERVED field of the TABLES table.

### Implementation

If the NLS bit is set for a table in TABLES.RESERVED, the table has its syntax C and V data represented in the system code page, which is always IBM-037. If the NLS bit is not set, the syntax C and V data of the table is represented in the TDS code page, which is set by the customer or database administrator.



- Support for subview (SUB) and calculation (CLC) tables could be affected – refer to [Subview and Calculation Tables on page 33](#) for further details.
- The value of the NLS bit does not affect processing on the local node. It is used to decide what translations are necessary in peer-to-peer communication.
- This use of the NLS bit applies to data parameters as well as data fields. Location parameters are always interpreted as being represented in the system code page.

### Tables with NLS Bit Pre-set

The NLS bit is set on all tables (with a few exceptions) that satisfy all three of the following conditions:

- The table is shipped with TIBCO Object Service Broker.
- The table type is TDS.
- The table contains syntax C or V fields or data parameters.

### Setting the NLS Bit

To turn the NLS bit on, that is, to indicate that NLS translation should not be used for this table, use

```
SETNLSBIT (tablename, 'Y')
```

To turn it off, use:

SETNLSBIT (*tablename*, 'N')



With the implementation of data translations for peer-to-peer communication, tables that have mixed usage could produce unexpected results in remote requests. Therefore, if you chose a TDS code page that is not the same as the system code page, non-English characters or special characters contained in the fields or parameters of a table with the NLS bit set could produce results different from what you expect.

## Expanded EBCDIC Code Page Substitution Characters

### Special Characters for TIBCO Object Service Broker Names

The characters listed in the table below are used in National EBCDIC systems (for example, z/OS) to represent the characters that must be entered in place of the special characters dollar sign (\$), number sign (#), and at sign (@) when used in TIBCO Object Service Broker object names, for example, rule or table names. These characters occupy the same code points in the indicated National EBCDIC code page as the special characters occupy in code page IBM-037.

#### Substitution Characters

Locale	Code Pages	Substitute for \$	Substitute for #	Substitute for @
CDNB	IBM-037, IBM-1140	\$	#	@
DANS	IBM-277, IBM-1142	Å	Æ	Ø
DEUT	IBM-273, IBM-1141	\$	#	§
ENGB	IBM-285, IBM-1146	£	#	@
ENGL	IBM-037, IBM-1140	\$	#	@
ESPA	IBM-284, IBM-1145	\$	Ñ	@
FRAN	IBM-297, IBM-1147	\$	£	á
NORS	IBM-277, IBM-1142	Å	Æ	Ø
ITAL	IBM-280, IBM-1144	\$	£	§
PORT	IBM-282	\$	Ã	Õ
SCHW	IBM-500, IBM-1148	\$	#	@
SUOM	IBM-278, IBM-1143	Å	Ä	Ö
SVEN	IBM-278, IBM-1143	Å	Ä	Ö

## Evaluation of TCP (TDS Code Page) versus SCP (System Code Page) Value

If the TIBCO Object Service Broker TDS code page value is specified as something other than the system code page (refer to [Chapter 6, Specification of NLS Characteristics – @NLS1 Table, on page 37](#) for information about the *locale.codepage* setting for SELF in the @NLS1 table), the code points for the dollar sign (\$), number sign (#), and at sign (@) characters can differ from those of the system code page. While these characters can be entered, they would occupy the appropriate code points for the TDS code page, not the correct code points for the system code page.

For example, for a TDS code page setting of NORS.IBM-277, a dollar sign (\$) in a customer data character string is represented as code point 0x67 in the TDS code page, whereas it is represented in code point 0x5B in the system code page (IBM-037). This interpretation of code page translations favors the entry and interpretation of customer data in its natural form at the expense of TIBCO Object Service Broker object names.

## Post-installation Peer NLS Testing

---

To verify the correct operation of TIBCO Object Service Broker in a peer environment, after valid values of the @NLS1 table are defined on each peer, perform the test described below.

### Verification Test

If there are two peer nodes, NODEA and NODEB, operating in a National EBCDIC NLS environment, perform the following:

1. Define a table on NODEA and NODEB, parameterized by LOCATION and containing three fields: a binary key, a type V character field of 16 characters, and a type C character field of 16 characters.
2. From each local node, populate the local table with accented characters entered from the locally attached keyboard.

Enter the same lowercase and uppercase characters in both V and C table fields.

3. Access the remote table and verify that the characters entered from NODEA are identical to those that appear on a remote request from NODEB, and repeat from NODEB, looking at the characters on NODEA.

Ensure the accented letters that are part of your locale are correctly folded to uppercase in the type C fields.

### Windows Behavior

Under Windows, certain keyboard characters could be entered incorrectly because Windows could be using them as shortcut keys for Windows-specific purposes. This is not a TIBCO Object Service Broker problem; it is a feature of the underlying Windows support.

## Chapter 8

# Configuration for the Solaris Platform

This chapter describes the setup needed to customize the NLS-enabled Execution Environment for the Solaris platform.

## Topics

---

- [Setup for the Solaris Platform, page 56](#)
- [Solaris Locale Support, page 58](#)
- [Testing Your NLS Environment, page 59](#)

## Setup for the Solaris Platform

---

This section defines the setup needed to customize the NLS-enabled Execution Environment for the Solaris platform. It identifies the configuration requirements for TIBCO Object Service Broker to perform under the Solaris system.

### Supported Interfaces

Currently, support is provided for:

- ASCII X-Windows servers running the ISO8859-1 code page using the ostty interface
- ASCII devices running the ISO8859-1 code page using the Curses line mode tty support with the ostty interface

### Enabling Operation

You must enable operation with the extended ASCII ISO8859-1 code page on your Solaris platform prior to running TIBCO Object Service Broker.



Failure to enable operation with the extended ASCII ISO8859-1 code page on the Solaris platform before running TIBCO Object Service Broker results in undefined TIBCO Object Service Broker behavior.

### Required Steps

Complete the following steps:

1. Ensure the proper X-Windows server, client, and tty support is present and configured, and that your keyboard support software is properly configured to match the keyboard layout you are using.
2. Verify platform support that is provided for your locale.
3. Run the system-specific shell commands to initialize your session for your locale.
4. Run ostty to execute TIBCO Object Service Broker.

Perform steps #1. and #2. only once during the setup process to enable NLS. Perform steps #3. and #4. each time you want to run TIBCO Object Service Broker in an NLS-enabled environment.



## X-Windows Support and Keyboard Support

Internationalization support is provided by X-Windows, which operates in the ISO8859-1 ASCII code page by default. This support was tested with an X-Windows server (for example, Exceed for Windows) and takes advantage of the code page key mapping support for keyboards with different key layouts.

For example, to configure a Canadian Bilingual keyboard using Exceed, select the Configuration option and the appropriate keyboard layout. You must then restart the X-Windows server to take advantage of the altered definition. Other X-servers can differ, so you could need to do some customization to properly configure the keyboard you use.

### Non-English Keyboard Considerations

The standard U.S. English keyboard has 101 keys and most other Latin-1 keyboards have 102 keys with differing layouts. Most X-Windows servers and clients are sensitive to the number of keys on the keyboard. Because of the different number of keys, unexpected results can occur if you try to configure a non-U.S. keyboard layout using a 101-key keyboard.

The X Mode Switch key is usually configured by the X-server and client to enable the third set of keyboard characters on the non-U.S. English keyboard.

The client side of the X-Windows support (the Solaris host) must also be enabled to accept the high-order ASCII characters contained in the ISO8859-1 character set.

## X-Client Support

The following command enables the Solaris platform to accept 8-bit ASCII characters from the terminal server:

```
stty -parity -istrip
```

In addition, the X-terminal program user ID on each platform must be aware of the internationalized locale that corresponds to the keyboard used. This is done by setting localization environment variables to string values that correspond to the system-specific locale name or its alias on each platform.

## Solaris Locale Support

---

### Customization

On Solaris, you can use commands similar to the following to customize the session for NLS support:

```
stty -parity -istrip
export LANG=""
export LC_COLLATE=C
export LC_CTYPE=da.ISO8859-15
export LC_MONETARY=C
export LC_NUMERIC=C
export LC_TIME=C
export LC_MESSAGES=C
export LC_ALL=""
xterm
```

### X-Windows Keyboard Support

Note the following:

- Some X-Windows emulators do not support a Finnish keyboard layout. In this case, the Swedish keyboard layout can be used.
- Some X-Windows emulators do not support a Danish keyboard layout. In this case, the Norwegian keyboard layout can be used.

## Testing Your NLS Environment

---

### Verification

Before attempting to run TIBCO Object Service Broker, you should verify that your system is responding to accented NLS characters as you expect.

From the Solaris command line, type some accented characters that are on your keyboard and are not part of the U.S. keyboard character set. For example, enter the e-acute character (é) on the French or Canadian Bilingual keyboards, or the umlaut character (ü) on the German keyboard. These characters should be echoed back correctly. If they do not appear correctly and you ran the appropriate shell script for your platform (as described in the previous section), ask your system administrator properly configure your setup.

### Testing your NLS-enabled TIBCO Object Service Broker on Solaris

After verifying that accented characters are properly echoed when entered from the command line, you can invoke TIBCO Object Service Broker clients.



## Appendix A **NLS Supported Code Pages**

This appendix describes the EBCDIC and ASCII code pages supported for NLS in TIBCO Object Service Broker.

### Topics

---

- [EBCDIC Code Pages, page 62](#)
- [ASCII Code Pages, page 68](#)

## EBCDIC Code Pages

---

The following EBCDIC code pages are supported by TIBCO Object Service Broker. They are taken from the Latin-1 (Western Europe) group of character sets.

### With Euro Symbol Support

The code pages that support the euro symbol (€) are listed here and are described at [www.ibm.com/servers/eserver/series/software/globalization/codepages.html](http://www.ibm.com/servers/eserver/series/software/globalization/codepages.html) :

- Code page IBM-1140 (ENGL, CDNB)
- Code page IBM-1141 (DEUT)
- Code page IBM-1142 (DANS, NORS)
- Code page IBM-1143 (SUOM, SVEN)
- Code page IBM-1144 (ITAL)
- Code page IBM-1145 (ESPA)
- Code page IBM-1146 (ENGB)
- Code page IBM-1147 (FRAN)
- Code page IBM-1148 (SCHW)

In these code pages, code point x'9F' represents the euro sign. TIBCO Object Service Broker uses code point x'20' to represent the universal currency symbol for translation purposes.

### Without Euro Symbol Support

The EBCDIC code pages that do not support the euro symbol are described in the sections below.

In these code pages, code point x'9F' represents the universal currency symbol. TIBCO Object Service Broker uses code point x'20' to represent the euro sign for translation purposes.

## Code Page IBM-037 (ENGL, CDNB)

	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0					sp	&	-	ø	Ø	°	μ	^	{	}	\	0
-1					rsp	é	/	É	a	j	~	£	A	J	+	1
-2					â	ê	Â	Ê	b	k	s	¥	B	K	S	2
-3					à	è	À	È	c	l	t	.	C	L	T	3
-4					à	è	À	È	d	m	u	©	D	M	U	4
-5					á	í	Á	Í	e	n	v	§	E	N	V	5
-6					â	î	Â	Î	f	o	w	¶	F	O	W	6
-7					ã	ï	Ã	Ï	g	p	x	¼	G	P	X	7
-8					ç	ì	Ç	Ì	h	q	y	½	H	Q	Y	8
-9					ñ	ß	Ñ	`	ı	r	z	¾	I	R	Z	9
-A					ø	ı	ı	:	«	»	ı	[	-	¹	²	³
-B					.	\$	,	#	»	°	¿	]	ó	ú	Ô	Û
-C					<	*	%	@	ð	æ	Ð	—	ó	ú	Ô	Û
-D					(	)	_	'	ý	,	Ý	-	ó	ú	Ô	Û
-E					+	;	>	=	þ	Æ	Þ	·	ó	ú	Ô	Û
-F						¬	?	"	±	α	®	×	ó	ý	Õ	

## Code Page IBM-273 (DEUT)

	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0					sp	&	-	ø	Ø	°	μ	ø	a	ü	Ö	0
-1					rsp	é	/	É	a	j	ß	£	A	J	+	1
-2					â	ê	Â	Ê	b	k	s	¥	B	K	S	2
-3					{	ë	[	Ë	c	l	t	.	C	L	T	3
-4					à	è	À	È	d	m	u	©	D	M	U	4
-5					á	í	Á	Í	e	n	v	@	E	N	V	5
-6					â	î	Â	Î	f	o	w	¶	F	O	W	6
-7					ã	ï	Ã	Ï	g	p	x	¼	G	P	X	7
-8					ç	ì	Ç	Ì	h	q	y	½	H	Q	Y	8
-9					ñ	~	Ñ	`	ı	r	z	¾	I	R	Z	9
-A					Ä	Ü	ö	:	«	»	ı	¬	-	¹	²	³
-B					.	\$	,	#	»	°	¿		ó	ú	Ô	Û
-C					<	*	%	§	ð	æ	Ð	—		}	\	]
-D					(	)	_	'	ý	,	Ý	-	ó	ú	Ô	Û
-E					+	;	>	=	þ	Æ	Þ	·	ó	ú	Ô	Û
-F						^	?	"	±	α	®	×	ó	ý	Õ	

Code Page IBM-277 (DANS, NORs)

	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0					sp	&	-		@	°	μ	ø	æ	å	\	0
-1					rsp	é	/	É	a	j	ü	£	A	J	+	1
-2					â	ê	Â	Ê	b	k	s	¥	B	K	S	2
-3					à	è	À	È	c	l	t	·	C	L	T	3
-4					â	è	À	È	d	m	u	©	D	M	U	4
-5					á	í	Á	Í	e	n	v	§	E	N	V	5
-6					â	î	Â	Î	f	o	w	¶	F	O	W	6
-7					}	ï	\$	İ	g	p	x	¼	G	P	X	7
-8					ç	ì	Ç	İ	h	q	y	½	H	Q	Y	8
-9					ñ	ß	Ñ	`	ı	r	z	¾	I	R	Z	9
-A					#	ø	ø	:	«	ª	ı	¬	-	¹	²	³
-B					.	Å	,	Æ	»	°	¿		ø	ù	Ô	Û
-C					<	*	%	Ø	ð	{	Ð	—	ø	~	Ö	Ü
-D					(	)	_	'	ý	,	Ý	ˆ	ø	ù	Ò	Ù
-E					+	;	>	=	þ	[	Þ	˙	ó	ú	Ó	Ú
-F					ı	^	?	"	±	]	®	×	ó	ý	Õ	

Code Page IBM-278 (SUOM, SVEN)

	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0					sp	&	-	ø	Ø	°	μ	ø	å	å	É	0
-1					rsp	`	/	\	a	j	ü	£	A	J	+	1
-2					â	ê	Â	Ê	b	k	s	¥	B	K	S	2
-3					{	ë	#	Ë	c	l	t	·	C	L	T	3
-4					à	è	À	È	d	m	u	©	D	M	U	4
-5					á	í	Á	Í	e	n	v	[	E	N	V	5
-6					â	î	Â	Î	f	o	w	¶	F	O	W	6
-7					}	ï	\$	İ	g	p	x	¼	G	P	X	7
-8					ç	ì	Ç	İ	h	q	y	½	H	Q	Y	8
-9					ñ	ß	Ñ	é	ı	r	z	¾	I	R	Z	9
-A					§	ø	ø	:	«	ª	ı	¬	-	¹	²	³
-B					.	Å	,	Ä	»	°	¿		ø	ù	Ô	Û
-C					<	*	%	Ö	ð	æ	Ð	—		~	@	Ü
-D					(	)	_	'	ý	,	Ý	ˆ	ø	ù	Ò	Ù
-E					+	;	>	=	þ	Æ	Þ	˙	ó	ú	Ó	Ú
-F					ı	^	?	"	±	]	®	×	ó	ý	Õ	



## Code Page IBM-280 (ITAL)

	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0					sp	&	-	ø	Ø	[	μ	ϕ	à	è	ç	0
-1					rsp	]	/	É	a	j	ì	#	A	J	+	1
-2					â	ê	Â	Ê	b	k	s	¥	B	K	S	2
-3					ä	ë	Ä	Ë	c	l	t	·	C	L	T	3
-4					{	}	À	È	d	m	u	©	D	M	U	4
-5					á	í	Á	Í	e	n	v	@	E	N	V	5
-6					â	î	Â	Î	f	o	w	¶	F	O	W	6
-7					ã	ï	Ã	Ï	g	p	x	¼	G	P	X	7
-8					\	~	Ç	ì	h	q	y	½	H	Q	Y	8
-9					ñ	ß	Ñ	ù	i	r	z	¾	I	R	Z	9
-A					°	é	ò	:	«	ª	¡	¬	-	¹	²	³
-B					.	\$	,	£	»	º	¿		ô	û	Ô	Û
-C					<	*	%	§	ð	æ	Ð	—	ó	ü	Ö	Ü
-D					(	)	_	'	ý	,	Ý	ˆ	ı	˘	Ò	Ù
-E					+	;	>	=	þ	Æ	Þ	˙	ó	ú	Ó	Ú
-F					ı	^	?	"	±	α	®	×	ô	ÿ	Õ	

## Code Page IBM-282 (PORT)

	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0					sp	&	-	ø	Ø	°	μ	ϕ	ã	˘	Ç	0
-1					rsp	é	/	É	a	j	ç	£	A	J	+	1
-2					â	ê	Â	Ê	b	k	s	¥	B	K	S	2
-3					ä	ë	Ä	Ë	c	l	t	·	C	L	T	3
-4					à	è	À	È	d	m	u	©	D	M	U	4
-5					á	í	Á	Í	e	n	v	§	E	N	V	5
-6					{	î	#	Î	f	o	w	¶	F	O	W	6
-7					ã	ï	Ã	Ï	g	p	x	¼	G	P	X	7
-8					~	ì	\	Ì	h	q	y	½	H	Q	Y	8
-9					ñ	ß	Ñ	˘	i	r	z	¾	I	R	Z	9
-A					[	]	ó	:	«	ª	¡	¬	-	¹	²	³
-B					.	\$	,	Ã	»	º	¿		ô	û	Ô	Û
-C					<	*	%	Õ	ð	æ	Ð	—	ó	ü	Ö	Ü
-D					(	)	_	'	ý	,	Ý	ˆ	ô	ù	Ò	Ù
-E					+	;	>	=	þ	Æ	Þ	}	ó	ú	Ó	Ú
-F					ı	^	?	"	±	α	®	×	ı	ÿ	@	

Code Page IBM-284 (ESPA)

	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0					sp	&	-	ø	Ø	°	μ	φ	{	}	\	0
-1					rsp	é	/	É	a	j	ˉ	£	A	J	+	1
-2					â	ê	Â	Ê	b	k	s	¥	B	K	S	2
-3					à	è	À	È	c	l	t	·	C	L	T	3
-4					â	è	À	È	d	m	u	©	D	M	U	4
-5					á	í	Á	Í	e	n	v	§	E	N	V	5
-6					â	î	Â	Î	f	o	w	¶	F	O	W	6
-7					ã	ï	Ã	Ï	g	p	x	¼	G	P	X	7
-8					ç	ì	Ç	Ì	h	q	y	½	H	Q	Y	8
-9					ı	ß	#	`	ı	r	z	¾	I	R	Z	9
-A					[	]	ñ	:	«	ª	ı	^	-	¹	²	³
-B					.	\$	,	Ñ	»	º	¿	ı	ô	ù	Ô	Ù
-C					<	*	%	@	ð	æ	Ð	ˉ	ó	ü	Ö	Ü
-D					(	)	_	'	ý	,	Ý	˘	ô	ù	Ò	Û
-E					+	:	>	=	þ	Æ	Þ	˙	ó	ú	Ó	Ú
-F						¬	?	"	±	α	®	×	ô	ý	Õ	

Code Page IBM-285 (ENGB)

	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0					sp	&	-	ø	Ø	°	μ	φ	{	}	\	0
-1					rsp	é	/	É	a	j	ˉ	[	A	J	+	1
-2					â	ê	Â	Ê	b	k	s	¥	B	K	S	2
-3					à	è	À	È	c	l	t	·	C	L	T	3
-4					â	è	À	È	d	m	u	©	D	M	U	4
-5					á	í	Á	Í	e	n	v	§	E	N	V	5
-6					â	î	Â	Î	f	o	w	¶	F	O	W	6
-7					ã	ï	Ã	Ï	g	p	x	¼	G	P	X	7
-8					ç	ì	Ç	Ì	h	q	y	½	H	Q	Y	8
-9					ñ	ß	Ñ	`	ı	r	z	¾	I	R	Z	9
-A					\$	ı		:	«	ª	ı	^	-	¹	²	³
-B					.	£	,	#	»	º	¿	]	ô	ù	Ô	Ù
-C					<	*	%	@	ð	æ	Ð	˘	ó	ü	Ö	Ü
-D					(	)	_	'	ý	,	Ý	˘	ô	ù	Ò	Û
-E					+	:	>	=	þ	Æ	Þ	˙	ó	ú	Ó	Ú
-F						¬	?	"	±	α	®	×	ô	ý	Õ	

## Code Page IBM-297 (FRAN)

	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0					sp	&	-	ø	Ø	[	`	ø	é	è	ç	0
-1					rsp	{	/	É	a	j	~	#	A	J	+	1
-2					â	ê	Â	Ê	b	k	s	¥	B	K	S	2
-3					à	è	À	È	c	l	t	.	C	L	T	3
-4					@	}	À	È	d	m	u	©	D	M	U	4
-5					á	í	Á	Í	e	n	v	]	E	N	V	5
-6					ã	î	Ã	Î	f	o	w	¶	F	O	W	6
-7					å	ï	Å	Ï	g	p	x	¼	G	P	X	7
-8					\	ì	Ç	Ì	h	q	y	½	H	Q	Y	8
-9					ñ	ß	Ñ	μ	ι	ρ	z	¾	I	R	Z	9
-A					°	§	ù	:	«	ª	ı	¬	-	¹	²	³
-B					.	\$	,	£	»	º	¿		ô	û	Ô	Û
-C					<	*	%	à	ð	æ	Ð	—	ö	ü	Ö	Ü
-D					(	)	_	'	ý	,	Ý	~	ó	ı	Ò	Ù
-E					+	;	>	=	þ	Æ	Þ	ˆ	ó	ú	Ó	Ú
-F					ı	^	?	"	±	α	®	×	ô	ÿ	Õ	

## Code Page IBM-500 (SCHW)

	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0					sp	&	-	ø	Ø	°	μ	ø	{	}	\	0
-1					rsp	é	/	É	a	j	~	£	A	J	+	1
-2					â	ê	Â	Ê	b	k	s	¥	B	K	S	2
-3					à	è	À	È	c	l	t	.	C	L	T	3
-4					á	é	À	È	d	m	u	©	D	M	U	4
-5					á	í	Á	Í	e	n	v	§	E	N	V	5
-6					ã	î	Ã	Î	f	o	w	¶	F	O	W	6
-7					å	ï	Å	Ï	g	p	x	¼	G	P	X	7
-8					ç	ì	Ç	Ì	h	q	y	½	H	Q	Y	8
-9					ñ	ß	Ñ	`	ι	ρ	z	¾	I	R	Z	9
-A					[	]	ı	:	«	ª	ı	¬	-	¹	²	³
-B					.	\$	,	#	»	º	¿		ô	û	Ô	Û
-C					<	*	%	@	ð	æ	Ð	—	ö	ü	Ö	Ü
-D					(	)	_	'	ý	,	Ý	ˆ	ó	ı	Ò	Ù
-E					+	;	>	=	þ	Æ	Þ	ˆ	ó	ú	Ó	Ú
-F					ı	^	?	"	±	α	®	×	ô	ÿ	Õ	

## ASCII Code Pages

---

The following extended ASCII code pages are supported by TIBCO Object Service Broker. They are taken from the Latin-1 (Western Europe) group of character sets.

### With Euro Symbol Support

The code pages that support the euro symbol (€) are listed here:

- Code page ISO-8859-15, described at [www.microsoft.com/globaldev/reference/iso/28605.htm](http://www.microsoft.com/globaldev/reference/iso/28605.htm)

This code page does not support the universal currency symbol. Code point x'A4' is defined as the euro sign.

- Code page WIN-1252, described at [www.microsoft.com/globaldev/reference/sbcs/1252.htm](http://www.microsoft.com/globaldev/reference/sbcs/1252.htm)

This code page supports the universal currency symbol at code point x'A4' and the euro sign at code point x'80'.

### Without Euro Symbol Support

The ASCII code page that does not support the euro symbol is shown in the section below and described at [www.microsoft.com/globaldev/reference/iso/28591.htm](http://www.microsoft.com/globaldev/reference/iso/28591.htm).

In this code page, code point x'A4' is defined as the universal currency symbol.

## Code Page ISO8859-1

	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0			sp	0	@	P	`	p			rsp	°	À	Ð	à	ð
-1			!	1	A	Q	a	q			¡	±	Á	Ñ	á	ñ
-2			"	2	B	R	b	r			¢	²	Â	Ò	â	ò
-3			#	3	C	S	c	s			£	³	Ã	Ó	ã	ó
-4			\$	4	D	T	d	t			¤	´	Ä	Ô	ä	ô
-5			%	5	E	U	e	u			¥	µ	Å	Ö	å	ö
-6			&	6	F	V	f	v			¦	¶	Æ	Ø	æ	ø
-7			'	7	G	W	g	w			§	·	Ç	×	ç	÷
-8			(	8	H	X	h	x			¨	,	È	Ø	è	ø
-9			)	9	I	Y	i	y			©	¹	É	Ù	é	ù
-A			*	:	J	Z	j	z			ª	º	Ê	Ú	ê	ú
-B			+	;	K	[	k	{			«	»	Ë	Û	ë	û
-C			,	<	L	\	l				¬	¼	Ì	Ü	ì	ü
-D			-	=	M	]	m	}			-	½	Í	Ý	í	ý
-E			.	>	N	^	n	~			®	¾	Î	Þ	î	þ
-F			/	?	O	_	o				¯	¿	Ï	ß	ï	ÿ



# Index

## Symbols

@NLS1 table  
 \$SETOPT tool [31](#)  
 activating values [39](#)  
 example values [40](#)  
 explanation of [38](#)  
 field parameters [38](#)  
 initialization of [28](#)  
 required occurrences [38](#)  
 specification requirements [38](#)  
 @SERVERCONFIG [11](#)  
 \$SETOPT tool [31](#)

## A

accented characters  
 in TIBCO Object Service Broker object names [34](#)  
 LOWERCASE tool mapping of [33](#)  
 support for [3](#)  
 use of [2](#)  
 ASCII  
 converting to EBCDIC [25](#)  
 mapping between EBCDIC and [18](#)  
 ASCII occurrence  
 functionality of [40](#)

## B

BATCHLOAD\_CARDS tool [23](#)  
 binary values in shaded table entries [17](#)

## C

calculation (CLC) tables  
 effects of NLS bit [50](#)  
 support of [33](#)  
 CECP (Country Extended Code Page), definition of [8](#)  
 character strings  
 effects of values for SELF code page [42](#)  
 produced by GENBIN tool [16](#)  
 treatment of non-character values in [16](#)  
 character support  
 character sets supported [8](#)  
 characters affected by collation [19](#)

## code pages

- as used with TDS table 4
- as used with X-Windows 56
- definition of 4
- EBCDIC 10
- IBM-037, example of 8, 63
- IBM-1140, example of 62
- IBM-1141, example of 62
- IBM-1142, example of 62
- IBM-1143, example of 62
- IBM-1144, example of 62
- IBM-1145, example of 62
- IBM-1146, example of 62
- IBM-1147, example of 62
- IBM-1148, example of 62
- IBM-273, example of 63
- IBM-277, example of 64
- IBM-278, example of 64
- IBM-280, example of 65
- IBM-282, example of 65
- IBM-284, example of 66
- IBM-285, example of 66
- IBM-297, example of 67
- IBM-500, example of 67
- ISO8859-1, example of 69
- ISO8859-15, example of 68
- mapping of binary values 18
- minimizing translation for remote accesses 46
- specifications 40
- supported 10
- system 4
- translation of shaded entries 17
- translation, interpretation of 53
- use in communications 4
- values for REMOTE 42
- values for SELF 42
- WIN-1252, example of 68

## code points

- control character representation 9
- definition of 8
- how to determine 8
- hyphen character representation 9
- invalid 9
- overview of 8
- shaded table entries 9
- space character representation 9
- use in peer-to-peer communications 16
- collating sequence, cause of differences in 19
- collation
  - behavior of 19
  - character behavior 19
  - determining sort order 19
  - effects of sort order 19
  - example of code point differences 19
  - inconsistencies in sorting behavior 19
  - sequence ramifications 19
- commands, Solaris customization 58
- control character code points 9
- converting from ASCII to EBCDIC 25
- Country Extended Code Page (CECP), definition of 8
- customer support xvi
- customization for Solaris 56, 57, 58

## D

## data translation

- conditions required for remote access 17
- enabling NLS 28
- for remote access 17
- in peer-to-peer communication 16
- non-character data 16
- requirements for peer-to-peer communication 16
- types of data values 16
- decimal separator, setting of 33
- determining locale.codepage specifications 46
- display font support
  - Solaris 30
  - Windows 29
  - z/OS 30



## E

### EBCDIC

- Country Extended Code Pages (CECP) [8](#)
- mapping between ASCII and [18](#)
- special characters in expanded set [52](#)
- supported code pages in [10](#)
- verification test for NLS [54](#)

### effects of

- NLS bit on syntax C and V [50](#)
- sort order on data requests [19](#)

### enabling NLS

- locale.codepage requirements [40](#)
- overview [40](#)
- tasks required for [28](#)

### euro symbol, support for [11](#)

### Execution Environment, code page used in [4](#)

- expanded character set, support requirements on Solaris [32](#)

## F

### field parameters for @NLS1 table [38](#)

### file transfer process

- audit log produced [24](#)
- BATCHLOAD\_CARDS tool [23](#)
- constraints and restrictions [25](#)
- File Transfer Protocol [24](#)
- hrnbrnls [23](#)
- hrnbrtbl utility [24](#)
- hrnbrula utility [22](#)
- hrnbrulh utility [22](#)
- hrnnbrulb utility [22](#)
- overview of [23](#)
- procedure [23](#)
- recommendations for sorting [24](#)
- requirements [22](#)
- S6BBRNLS [23](#)
- S6BBRTBL utility [24](#)
- S6BBRULA utility [22](#)
- S6BBRULH utility [22](#)
- S6BNBRULB utility [22](#)
- sorting considerations [24](#)

- folding rules, effects of valid locale values on [5](#)

## G

### GENBIN tool, character strings produced by [16](#)

## H

### hrnbrnls utility

- functionality of [22](#)
- use in converting from ASCII to EBCDIC [25](#)

### hrnbrtbl utility [24](#)

### hrnbrula utility [22](#)

### hrnbrulb utility [22](#)

### hrnbrulh utility [22](#)

### hyphen character code points [9](#)

## I

### IBM-037 code page, example of [8](#), [63](#)

### IBM-1140 code page, example of [62](#)

### IBM-1141 code page, example of [62](#)

### IBM-1142 code page, example of [62](#)

### IBM-1143 code page, example of [62](#)

### IBM-1144 code page, example of [62](#)

### IBM-1145 code page, example of [62](#)

### IBM-1146 code page, example of [62](#)

### IBM-1147 code page, example of [62](#)

### IBM-1148 code page, example of [62](#)

### IBM-273 code page, example of [63](#)

### IBM-277 code page, example of [64](#)

### IBM-278 code page, example of [64](#)

### IBM-280 code page, example of [65](#)

### IBM-282 code page, example of [65](#)

### IBM-284 code page, example of [66](#)

### IBM-285 code page, example of [66](#)

### IBM-297 code page, example of [67](#)

### IBM-500 code page, example of [67](#)

### identifying a code point [8](#)

- interpretation of code page translations [53](#)
- invalid code points [9](#)
- ISO8859-1 code page
  - as used on Solaris [56](#)
  - ASCII to EBCDIC translation, remapping of [10](#)
  - support for ASCII [10](#)
  - TIBCO Object Service Broker use of [10](#)
- ISO8859-1 code page, example of [69](#)
- ISO8859-15 code page, example of [68](#)

## K

- keyboard support
  - 3270 emulator [29](#)
  - exploitation dependencies [28](#)
  - non-English keyboard on Solaris [57](#)
  - Solaris [29](#), [57](#)
  - Windows [28](#)
  - z/OS [29](#)

## L

- LOAD tool (interactive and batch) [34](#)
- locale
  - and code page, functionality of [4](#)
  - definition of [4](#)
  - in X-Client support, role in [57](#)
  - specifications [40](#)
  - specifications, effects of [4](#)
  - values for REMOTE [43](#)
  - values for SELF [42](#)
- locale.codepage requirements [40](#)
- locale.codepage specifications
  - determining [46](#)
  - examples of [46](#)
  - valid values for [44](#)
- LOWERCASE tool [33](#)

## M

- mapping
  - accented characters with LOWERCASE tool [33](#)
  - binary values [18](#)
  - shaded table entries, explanation of [17](#)
  - valid locale values [5](#)
- minimizing necessity of code page translation [46](#)

## N

- naming restrictions in an NLS-enabled environment [34](#), [34](#)
- National Language Support (NLS)
  - advantages of [2](#)
  - definition of [2](#)
  - display font support [29](#)
  - effect on user IDs and passwords [34](#)
  - effect on userids and passwords [35](#)
  - enabling [28](#), [40](#)
  - functionality of [4](#)
  - keyboard support [29](#)
  - node name conventions [34](#), [34](#)
  - overview of configuration for [38](#)
  - scope of support [2](#)
  - setup and configuration of [28](#)
  - setup requirements for Solaris [56](#)
  - setup verification [54](#)
  - support of accented characters [3](#)
  - use of accented characters [2](#)
  - use of characters in data [2](#)
- NLS bit
  - effects on TDS code page [50](#)
  - implementation [50](#)
  - overview of [50](#)
  - role in peer-to-peer communication [50](#)
  - tables having mixed usage [51](#)
  - tables pre-set with [50](#)
- NLS configuration, overview of [38](#)
- NLS customization for Solaris [56](#), [58](#)
- Non-character data translation [16](#)

## P

- passwords, NLS restrictions [34, 35](#)
- peer data requests, support of [33](#)
- peer-to-peer communication
  - data translation in [16](#)
  - effects of tables having mixed usage [51](#)
  - enabling NLS [28](#)
  - example of [17](#)
  - role of NLS bit in [50](#)
  - translation requirements [16](#)

## R

- remote access
  - translation for [17](#)
  - translation requirements [17](#)
- REMOTE occurrence
  - @NLS1 table [38](#)
  - CompType field [38](#)
  - definition of [4](#)
  - functionality of [40](#)
  - values for code page [42](#)
  - values for locale [43](#)
- requirements for file transfer process [22](#)
- revising NLS settings [39](#)

## S

- S6BBRNLS utility
  - functionality of [22](#)
  - use in converting from ASCII to EBCDIC [25](#)
- S6BBRTBL utility [24](#)
- S6BBRULA utility [22](#)
- S6BBRULB utility [22](#)
- S6BBRULH utility [22](#)

- SELF occurrence
  - @NLS1 table [38](#)
  - CompType field [38](#)
  - definition of [4](#)
  - functionality of [40](#)
  - values for code page [42](#)
  - values for locale [42](#)
- setting the decimal separator [33](#)
- setup and configuration of NLS [28](#)
- shaded table entries
  - code points [9](#)
  - explanation of mapping for [17](#)
  - specification of binary values [17](#)
- Solaris
  - customization for NLS support [56, 57, 58](#)
  - display font support [30](#)
  - enabling NLS [56](#)
  - keyboard support [29, 57](#)
  - non-English keyboard considerations [57](#)
  - support for syntax W data items [34](#)
  - support of expanded character set [32](#)
  - supported interfaces [56](#)
  - valid code pages for SELF [42](#)
  - verification of NLS setup [59](#)
  - X-Client support [57](#)
  - X-Windows support [57](#)
- sort order in collation [19](#)
- sorting considerations for file transfers [24](#)
- space character code points [9](#)
- special characters in TIBCO Object Service Broker
  - object names, use of [52](#)
- subview (SUB) tables
  - effects of NLS bit [50](#)
  - support of [33](#)
- support, contacting [xvi](#)
- supported code pages [10](#)
- syntax C and V data, effects of NLS bit on [50](#)
- syntax W data items, support for [34](#)
- system tables, support of peer data requests for [33](#)

## T

- table definitions, transferring between Data Object

- Brokers [33](#)
- TABLES.RESERVED, NLS bit in [50](#)
- TDS code pages
  - default value used for [8](#)
  - effect on table definitions [33](#)
  - effects of NLS bit [50](#)
  - LOAD and UNLOAD tools (interactive and batch) [34](#)
  - relationship with system code page [53](#)
  - valid [11](#)
- TDS tables
  - and NLS bit [50](#)
  - code pages associated with [4](#)
- technical support [xvi](#)
- testing NLS-enabled TIBCO Object Service Broker on Solaris [59](#)
- TIBCO Object Service Broker object names
  - accented characters in [34](#)
  - special characters in [52](#)
- TIBCO\_HOME [xiii](#)
- tools
  - \$SETOPT [31](#)
  - BATCHLOAD\_CARDS [23](#)
  - GENBIN [16](#)
  - LOAD (interactive and batch) [34](#)
  - LOWERCASE [33](#)
  - UNLOAD (interactive and batch) [34](#)
- types of data translated [16](#)

## U

- unbinding @NLS1 table [39](#)
- UNLOAD tool (interactive and batch) [34](#)
- use of NLS bit [50](#)
- user IDs, NLS restrictions for [34](#)
- user-defined tables, support of peer data requests for [33](#)
- userids, NLS restrictions for [35](#)

- utilities
  - hrnbrnls [22](#)
  - hrnbrtbl [24](#)
  - hrnbrula [22](#)
  - hrnbrulb [22](#)
  - hrnbrulh [22](#)
  - S6BBRNLS [22](#)
  - S6BBRTBL [24](#)
  - S6BBRULA [22](#)
  - S6BBRULB [22](#)
  - S6BBRULH [22](#)

## V

- valid locale.codepage specifications [44](#)
- valid TDS code pages [11](#)
- verification of NLS setup on Solaris [59](#)
- verifying setup of NLS-enabled TIBCO Object Service Broker [54](#)

## W

- WIN-1252 code page, example of [68](#)
- Windows
  - display font support [29](#)
  - keyboard behavior [54](#)
  - keyboard support [28](#)
  - support for syntax W Data items [34](#)
  - valid code pages for SELF [42](#)

## X

- X-Client support [57](#)
- X-Windows
  - keyboard considerations [58](#)
  - support for [57](#)

## Z

### z/OS

- 3270 emulator and keyboard support [29](#)
- changing the TDS code page [31](#)
- display font support [30](#)
- keyboard support [29](#)
- special characters in TIBCO Object Service Broker
  - object names [52](#)