

Understanding Unicode

in Sage SalesLogix Version 8.0

Developed by Sage SalesLogix User Assistance



sage

Copyright © 1997-2012, Sage Software, Inc. All Rights Reserved.

This product and related documentation are protected by copyright and are distributed under licenses restricting their use, copying, distribution, and decompilation. No part of this product or related documentation may be reproduced in any form by any means without prior written authorization of Sage and its licensors, if any.

Version 8.0 (080312)

Sage has thoroughly reviewed this manual. All statements, technical information, and recommendations in this manual and in any guides or related documents are believed reliable, but the accuracy and completeness thereof are not guaranteed or warranted, and they are not intended to be, nor should they be understood to be, representations or warranties concerning the products described. Sage assumes no responsibility or liability for errors or inaccuracies with respect to this publication or usage of information. Further, Sage reserves the right to make changes to the information described in this manual at any time without notice and without obligation to notify any person of such changes.

Sage

8800 N. Gainey Center Dr., Suite 200

Scottsdale, AZ 85258

<http://na.sage.com/crm/> | 800-854-3415

What is Unicode?	1
Understanding Unicode in Sage SalesLogix	1
Enabling Unicode	3
Adding Unicode Fields to Tables	4
Converting a Field to Unicode	5
Modifying Fields.....	6
Unsupported Tables.....	6
Troubleshooting Unicode	7

What is Unicode?

Unicode is an international 16-bit character set and encoding system designed to represent all the characters of all the world's languages. The actual character is not stored, just a byte value or ID which refers to a particular character. The character itself exists in the font set used on the local machine.

Understanding Unicode in Sage SalesLogix

The Sage SalesLogix database, Web applications, and Network Client support field-selectable Unicode. You can enable field-selectable Unicode and configure the system to allow targeted fields to be Unicode compliant. This allows you to create, view, update or delete records that are represented using a Unicode character set.

BENEFITS

When you enable field-selectable Unicode:

- Offline Web Client users can synchronize Unicode data in their remote database to the Host database.
- Clients can view multiple, multi-byte languages. For example, a user can work on an account that has been stored in Russian, then open an account that is stored in Japanese, and view the characters for each language correctly in the Account Detail view.
- Only one Sync Server is required for all languages. Previous multi-byte installations required one Sync Server for each language.

RISKS, LIMITATIONS, AND CONSIDERATIONS

Field-selectable Unicode is limited to specific feature areas. Extending the database for Unicode requires a strong knowledge of Sage SalesLogix and its database schema. Ensure you understand the risks and limitations of Unicode in Sage SalesLogix before enabling this feature.

- **Multi-byte data** - Do not enable field-selectable Unicode if your database contains multi-byte data. Multi-byte data is used for languages that require more than one byte of code to store a character. These are typically languages which use a pictograph or hieroglyphic symbol to convey meaning, like Chinese. Converting existing multi-byte data fields to Unicode may result in a loss or corruption of data.
- **Propagating tables** - When you convert a field to Unicode, you must change the same field in all propagating tables. If you do not change all related entities, data will be corrupted and display incorrectly. For example, if the Account field is used in the Account table and the Account_Extra custom table, the field must be converted (set) in the Account_Extra table as well as the Account table.
- **Pick lists** - Any field that is populated by a Unicode pick list must have Unicode enabled as well.
- **Upgrades** - Converting fields to Unicode is considered a customization for upgrade purposes because it is a change to the entity model; upgrades after you convert to Unicode may require that you bundle the Unicode model and merge it.

- **Unsupported applications** - The following areas in Sage SalesLogix will not work with Unicode data.



If your database contains Unicode data and you continue to use these applications, unexpected results may occur, including the loss or corruption of data, which may not be recoverable.

Application/Functionality	With Unicode Enabled
E-mail	E-mail addresses do not support Unicode due to a limitation of the email protocol. You cannot use an e-mail address that contains Unicode characters (the SMTP standard), but the display name can contain Unicode characters.
ExchangeLink	Use ExchangeLink v8.0 SP1 or later. Prior versions of ExchangeLink do not work with Unicode data.
GroupCopy	Does not work with Unicode data.
Import Wizard	Does not work with Unicode data. If you import Unicode data, it will be corrupted.
Integrity Checker	Does not work with Unicode data.
Administrator	Does not work with Unicode data.
Architect	Does not work with Unicode data.
Mail Merge on the Sage SalesLogix Network Client	Does not work with Unicode data.
SLXMail	Does not work with Unicode data.
Web Reporting	Works with limitations. If you have a report that filters by a group condition that contains Unicode, the report will not display correctly.

- **Unsupported functionality** - When implementing field-selectable Unicode, there are specific areas in Sage SalesLogix that cannot be modified.
 - Some tables do not support Unicode.
 - Groups with IN clauses cannot use fields in the IN clause that have been converted to Unicode. IN clauses only support literal string values. For example:

This statement will work: AccountID IN (SELECT ACCOUNTID FROM OPPORTUNITY WHERE SALESPOTENTIAL > 10000)

This statement will not: AccountID IN (SELECT ACCOUNTID FROM OPPORTUNITY WHERE DESCRIPTION LIKE '??%') - where ??% represents Unicode characters.

Enabling Unicode

Do not enable field-selectable Unicode until you have reviewed the risks and limitations associated with this feature. Do not enable Unicode if your database contains multi-byte data. Converting existing multi-byte data fields to Unicode may result in a loss or corruption of data.

When field-selectable Unicode is enabled:

- Unicode field types (Unicode String and Unicode Memo) become available in the Database Manager and Application Architect Entity Designer. Unicode String displays Unicode text on the clients.
- GroupTranslator sets conditions to UTF-16.
- Sage SalesLogix uses a new code page for ANSI String and UTF16.
- The System Summary Report is updated to indicate Unicode is enabled.
- Database storage requirements may increase. Ensure your main office and Remote user computers have sufficient hardware to store and process Unicode data.



Field-selectable Unicode is limited to specific feature areas. Extending the database for Unicode requires a strong knowledge of Sage SalesLogix and its database schema.



If you convert an existing field to Unicode or create a new Unicode field and write data to your database, you cannot revert your installation to ANSI data types. You can enable/disable Unicode and create/remove new Unicode fields *only* until you convert data or write Unicode data to your database. Once Unicode data has been written to the database, you will corrupt your data if you disable the changes.

To enable

1. Use Administrator to do the following:
 - a. On the **Tools** menu, click **Options**.
 - b. Click the **Database** tab.
 - c. Select the **Allow Unicode field types** check box.
 - d. Read the warning message, and then click **Yes**.
 - e. Click **OK**.
2. (Microsoft SQL Server Only) Disable auto-translate on your database server.
 - a. Open the Connection Manager.
 - b. In the **SalesLogix Connection Manager** dialog box, select your database connection and click **Edit**.
 - c. Click the **All** tab.
 - d. Select the **Auto Translate** property and click **Edit Value**.
 - e. In the **Property Value** box, select **False**.
 - f. Click **OK**.
 - g. Save your settings and exit the Connection Manager.
 - h. Restart the server.
3. (Oracle Only) If you are creating a new database instance, use AL32UTF8 as the Normal character set.
4. (Windows 2003/XP only) Configure supplemental language support on Web Host and Web Client machines if they must support East Asian or other multi-byte character sets:
 - a. In the Control Panel, open **Regional and Language Options**.
 - b. On the **Regional Options** and **Advanced** tabs, verify the correct language is selected in the drop-down list.

- c. If your required language is not available in the list, click the **Languages** tab and select the boxes for Supplemental language support. For example, if Chinese, Japanese, or Korean languages will be used or stored in the database, check **Install files for East Asian languages**. A file copy and restart may be required. When finished, select the appropriate language on the Regional Options and Advanced tabs and restart.
5. If you anticipate that your database will continue to contain some multi-byte data, then verify that the **Language for non-Unicode programs** value is defined on the Web Host and Web Client machines. This setting is used to determine the code page that is used to convert between multi-byte string data and Unicode data.
 - a. In the Control Panel, open **Regional and Language Options**.
 - b. On the **Advanced** tab (Windows XP/2003) or the **Administrative** tab (Windows 2008/7), select the appropriate language.
6. (Microsoft Office 2007 only) Enable UTF-8 support for Internet Protocols in Microsoft Outlook. UTF-8 support must be enabled when users are exporting extended characters to e-mail.
 - a. Open Microsoft Outlook.
 - b. On the **Tools** menu, click **Options**.
 - c. Click the **Mail Format** tab.
 - d. Click **International Options**.
 - e. Under **Internet Protocols**, select **Enable UTF-8 support for mailto: protocol**.
 - f. Click **OK**.

Adding Unicode Fields to Tables

You can create a custom table that includes Unicode fields, or add Unicode fields to an existing table.



Back up your database before making any Unicode changes.

To add a custom table that includes Unicode fields

1. Open the Database Manager.
2. Add a table and fields to your database.
3. Open the Application Architect.
4. Use the **New Entity Wizard** to create a new entity that points to your table.

Ensure you select the Create a business entity from an existing table option in the wizard. See the "Using the New Entity Wizard" topic in the Application Architect Help for details.

To add a Unicode field to an existing table

1. Open the Database Manager.
2. Add a field(s) to the table.
3. In Application Architect **Project Explorer > Entity Model > Packages**, right-click the existing entity to which you want to add the field.
4. Click **Update Properties**, select the field name(s), and then click **Finish**.

Converting a Field to Unicode

Use the following steps to convert an existing field to Unicode. Only fields of data type Text/String or Memo (and user-defined types based on them) can be converted to Unicode.



The information and procedures described in this document apply to a Microsoft SQL Server or an Oracle environment using a Latin1-based character set. If your Oracle database character set is already set to Unicode (UTF8, UTF16), you do not have to manually convert fields to Unicode, because the database is already set as Unicode.



- Back up your database before making any Unicode changes.
- If you convert a field to Unicode, do not convert it back. Understand all changes you want to make before implementing them on your database.
- Keep in mind that some fields such as primary and foreign keys will not work with Unicode data even if they have been converted to a Unicode data type.
- Do not convert a field that contains multi-byte data. Converting existing multi-byte data fields to Unicode may result in a loss or corruption of data.

To convert

1. Open your Microsoft SQL Server or Oracle database management tool.
You cannot use the Database Manager to convert fields to Unicode.
2. Modify the appropriate fields to Unicode data types.
 - Text/String field - change VarChar to nVarChar for Microsoft SQL Server and VarChar 2 to nVarChar 2 for Oracle.
 - Memo field - change Text to nText for Microsoft SQL Server and CLOB to nCLOB for Oracle.
3. Open the Application Architect.
4. In **Project Explorer > Entity Model > Packages**, double-click the entity that contains the fields you want to modify.
5. Click the field's **Data Type** to open the **System Data Type Editor**.
6. In the **Select Data Type** box, select the data type.
In the Application Architect, Memo fields are listed as Text. These Text fields map to a Memo field in the database and display correctly as Memo in the Database Manager. If the database type for a property is Text/Memo, the recommended Unicode type is Unicode Memo.

- Unicode Text - use for variable-length Unicode text.
- Unicode Memo - use for unlimited-length Unicode text.

7. Click **OK**.
8. Continue for all fields you want to convert. Ensure you change the same field in all propagating tables.
9. If your installation includes remotes, you must create new remote databases and distribute to all Remote users and Remote Offices.

When you modify a field outside the Database Manager, the changes are not logged by the Sage SalesLogix provider. To distribute the changes to remotes, you must create new remote databases.

Modifying Fields

When you convert a field to Unicode, you must change the same field in all propagating tables. If you do not change all related entities, data will be corrupted and display incorrectly.

For example, if you change the Account.AccountName field to Unicode but do not change the Contact.AccountName field, the AccountName field for the contact will display "???" in the Web Client. If your data is corrupted during this process, convert the fields in propagating tables that were not modified correctly. Then use a script to repopulate your data.

Unsupported Tables

The following is a list of tables for which Sage SalesLogix does not support Unicode data. Ensure your conversion plans do not include or impact these tables.

ADMINROLES	META_CLASS	SITEKEYS
AGENTS	META_FUNCTION	SITEOPTIONS
BRANCHOPTIONS	MM_MAINTABLE	SLXBUNDLE
CALCULATEDFIELDSDATA	MM_MAINTABLE_FIELD	SLXOLEDBPLUGIN
CONTROLCLASSES	MM_MAINTABLE_REL	SLXOLEDBPLUGINDATA
CUSTOMSETTINGS	NETFORMS_LICENSEKEY	SLXQUERYPLAN
DASHBOARDOPTIONS	NETFORMS_PLUGIN	SLXQUERYPLANHINT
EXCHANGERATE	NOTIFICATIONCONFIG	SLXSQLDEFAULT
FEATURESECURITY	PROJECT	SLXSQLSCALARFN
GM_PROFILE	PROJECTITEM	SLXTRIGGERS
INDEXDEFINITION	REALIGNPROFILE	SLXWEBUSERINFO
INDEXGENERAL	REMOTEAUTOCLEANUP	SUBSCRIPTIONRULES
INDEXSCHEDULE	REMOTETASKS	SYNCJOBHISTORY
INDEXSTATS	REPORTFILTER	SYNCSECTABLES
INDEXUPDATES	RESYNCTABLEDEFS	SYNCSERVER
JOINDATA	SALESDASHBOARD	SYNCSERVICEFILE
LANGUAGETRANSLATION	SECFUNCTIONS	SYNCTRANSFER
LEAD_IMPORTMAP	SECPROFILE	SYSTEMINFO

LICENSES	SECTABLEDEFS	USERFEATURESECURITY
LOOKUP	SITE	USERSECURITY
		WORKAREA

Troubleshooting Unicode

You may have questions on or encounter the following issues while working with Unicode:

- Characters do not look correct in Mail Merge.
Mail Merge does not support Unicode characters.
- How do I convert a Varchar field in Microsoft SQL Server to an nVarchar field?
To convert VarChar to nVarChar, use a SQL statement executed in the SQL Studio Manager. For example:
Alter table sysdba.account alter column userfield1 nvarchar(80)
- I want users to select pick list items that are in Unicode characters.
Pick lists can support Unicode characters. However, any field that is populated by a Unicode pick list must have Unicode enabled as well.
- Am I limited to using an nVarChar type field in Microsoft SQL Server?
You can use nVarChar, nChar, or nText.
- I am seeing question marks instead of my data in a field.
 - Make sure Unicode support is enabled and that any code that inserts Unicode data uses a query parameter (for example, OleDbParameter) to set the value; it cannot be set using a string literal because SLXOLEDB does not support the N prefix.

