SIL Encoding Converters

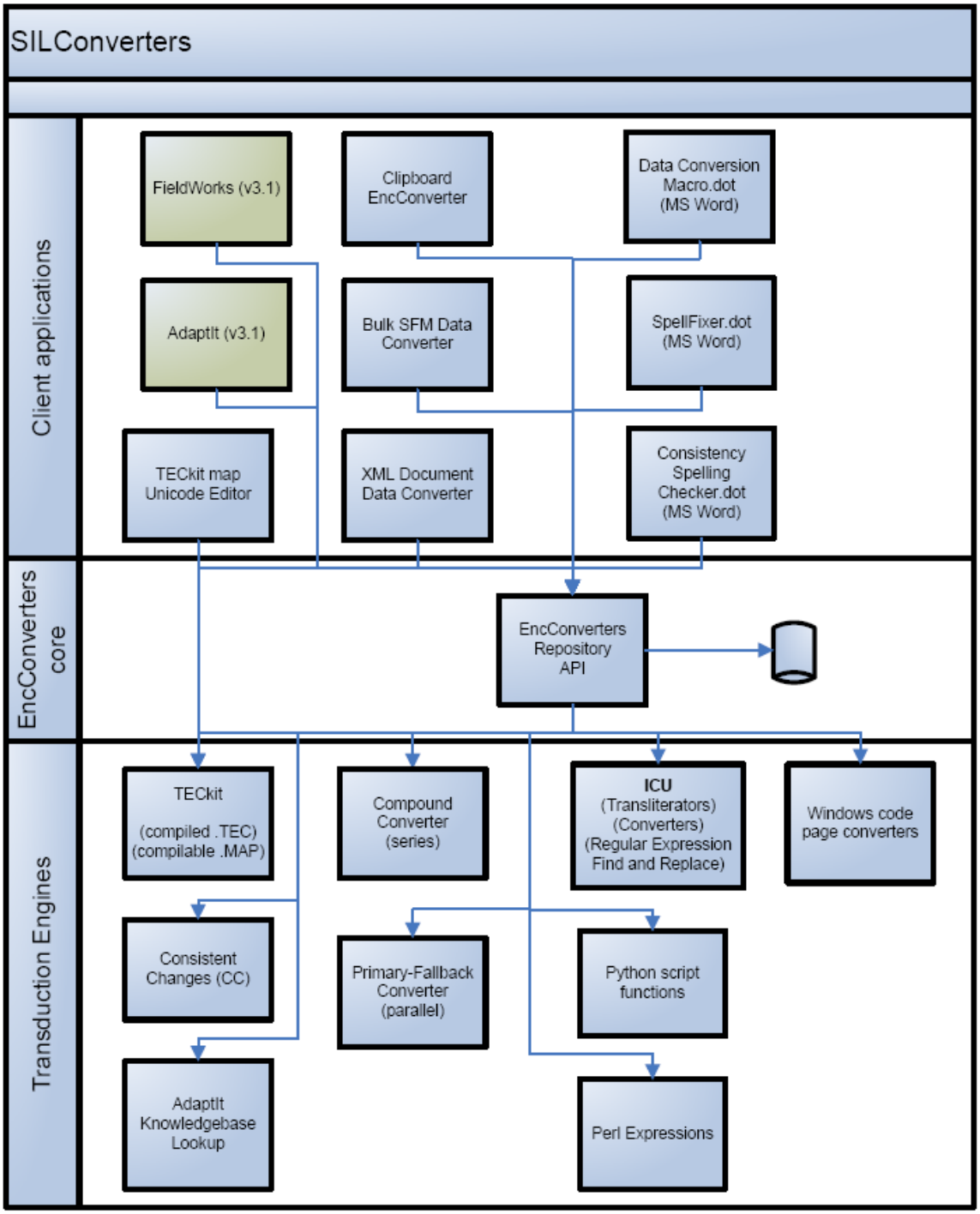
Ken Zook

April 1, 2011

Contents

1 Introduction

SIL Encoding Converters is a set of programs developed and supported by Bob Eaton for dealing with conversion between legacy encodings and Unicode, and vice versa. It also provides for various transducers for converting Unicode data to some other form of Unicode data. For detailed information on this program, see "Help for SIL Converters.htm" after unzipping SilConvertersHelp.zip.



This diagram summarizes the use of SIL Encoding Converters. The core program is a DLL (SilEncConverters30.dll) that gets installed in the Windows Global Assembly Cache (GAC). It maintains a repository of converters and transducers, usually in %ALLUSERSPROFILE%\Application Data\SIL\Repository, where %ALLUSERSPROFILE%\Application Data is c:\Documents and Settings\All Users\Application Data on Windows XP and c:\ProgramData on Vista. SIL Encoding Converters makes use of various engines to do actual conversion or transduction. Engines include Consistent Changes, TECkit, ICU, Python, Perl, Windows code pages, and the

ability to chain any other converters to produce a compound converter. Various applications can make use of these converters for import, export, or transduction of data. At this point FieldWorks and AdaptIt both use this system. Bob also provides a clipboard converter, a TECkit map editor, and several Word macros to work with the package as well.

The core functionality is installed as part of a FieldWorks installation, or users can download and install from the NRSI website at http://scripts.sil.org/cms/scripts/page.php?site_id=nrsi&cat_id=ConversionUtilities. The FieldWorks master installer allows users to install additional SIL Encoding Converter programs beyond the core functionality.

1.1 Alternatives to install converters

A. FieldWorks provides a dialog to manage most converters. You can access them in alternate ways:
   - Use RunAddConverters.exe (a standalone program in your FieldWorks directory) to access this dialog directly.
     Note: The Help button currently does not work in this mode.
   - Access any FieldWorks application through File…Project Management…FieldWorks Project Properties…Writing Systems.
     Tip: Flex provides a shortcut to this via Format…Setup Writing Systems.
     1. Select a writing system and then click Modify.
     2. Click Converters and then click More (next to Encoding converters).
        Tip: This is also available through the new Writing System Wizard.
   - Go to File…Import…Standard Format Lexicon. In Step 3, click Add or Modify and then click Add (next to the Encoding Converter combo).
   - Use File…Import…LinguaLinks data. Click Specify for a language definition, and then click Add (next to the Encoding Converter combo).

B. In a Flex Bulk Edit mode, click Process and then click Setup.
   Tip: This dialog provides a simplified way to use or set up Unicode to Unicode transducers.

C. The Clipboard Converter program comes with the SIL Encoding Converter package. Right-click the running icon and choose Add Converter.

D. The Data Conversion Macro 0300.dot Word macro comes with the SIL Encoding Converter package (see information below to install and use).

1.2 Version Compatibility Issues

SIL Encoding Converters 3.0.0 is available from the NRSI site, and the core parts are installed by FieldWorks 5.4. SIL Encoding Converters 3.0.0 now uses an XML file for storing some of the installation information that used to be stored in the registry. As a result, you can safely install and uninstall SIL Encoding Converters 3.0.0, FieldWorks 5.4, and Speech Analyzer 3.0.1 without breaking any of the applications.

History: Speech Analyzer 3.0.1 and FieldWorks 4.9 through 5.2 installed and used SIL Encoding Converters 2.6.1. There was a bug with registry keys during uninstallation of pre-3.0.0 releases that damaged some of the registry keys used by the other version. If you are working with these older versions, the safest thing is to install new versions of

each of these applications which will install and uninstall without affecting the other apps. If this is not possible, and you broke one application by uninstalling the other one, the problem can be fixed by re-registering the remaining SilEncConverters22.dll file. One way to do this is to store the following batch commands in a .bat file and executing that batch file.

```
@if "%_echo%"=="" echo off

set DLL_PATH260=%CommonProgramFiles%\SIL\2.6.0.0\SilEncConverters22.dll
set DLL_PATH261=%CommonProgramFiles%\SIL\2.6.1.0\SilEncConverters22.dll
set
REGASM_PATH=%SystemRoot%\Microsoft.NET\Framework\v2.0.50727\regasm.exe
set DID_WORK="no"

if not exist "%REGASM_PATH%" goto INVALID_REGASM

if not exist "%DLL_PATH260%" goto NO260_DLL
echo Registering SilEncConverters22.dll ver 2.6.0.0
%REGASM_PATH% "%DLL_PATH260%"
set DID_WORK="yes"

:NO260_DLL
if not exist "%DLL_PATH261%" goto DID_WORK
echo Registering SilEncConverters22.dll ver 2.6.1.0
%REGASM_PATH% "%DLL_PATH261%"
set DID_WORK="yes"

:DID_WORK
if %DID_WORK%=="no" echo ** Couldn't find either
SilEncConverters22.dll: 2.6.1.0 or 2.6.0.0
goto END

:INVALID_REGASM
echo ** RegAsm is not found : <%REGASM_PATH%>
goto END

:INVALID_DLL
echo ** SilEncConverters22.dll is not found : <%DLL_PATH260%>

:END
```

2 Using the FieldWorks Encoding Converters dialog

You can select an existing converter in the Available Converters list.

You can delete a converter from the repository by selecting it in the Available Converters list, then click Delete.

2.1 Adding a new converter

1. Click Add
2. Type a new name in the Converter Name edit box.
3. Select the desired converter engine from the Converter Type combo.
   Note: If the type of converter you want is not available in this list, click the More button to use the full SIL Encoding Converter installation dialog. From this dialog

you can add Perl, Python, Compound converters, etc. If you define a converter in this dialog, the following two steps will become unavailable since they are defined in the other dialog.

4. Depending on the type of converter, the line varies:
   A. Select from a fixed list.
   B. Select an external CC table or TECkit MAP (source) or TEC (compiled) file.
5. Select the desired Conversion Type.
   - For CC and ICU legacy converters this is usually Legacy to Unicode.
   - For TECkit and Windows code pages it is typically Legacy to and from Unicode.
   - For ICU transducers or some CC tables, it is Unicode to Unicode.

The Advanced tab gives some information about the selected converter.

2.2 Testing a converter

The Test tab provides a convenient way to test a converter on a data file. It is usually best to provide a relatively short test file that contains samples of the code points you want to convert.

1. Select a converter from the Available Converters list.
2. Click Select Input File and choose a file you want to convert.
3. Select an Output font that will display the converted Unicode characters.
4. Click Convert.
   Result: Converts the file and shows the results in the Converted pane.
   Tips: The specified converter applies to the *entire* file. If you have a standard format file where some fields use one converter while other fields use another converter, *only* the fields needing this converter will display correctly. This window is Graphite-enabled.
5. To save the converted results, click Save to File.

Ansi test.txt is a sample file you might find useful for testing conversion from a legacy font. It contains all the code points above space, so you can see what happens to each one using a given converter.

If you get a BroadcastEventWindow error message when closing RunAddConverters, ignore it.

3 Consistent changes

Use a Consistent Changes (CC) table for legacy-to-Unicode conversion and for Unicode-to-Unicode conversion (see Bulk edit issues.doc). CC will only process Unicode data in UTF-8 format. A CC table is *always* a one-way converter (e.g., the same table cannot be used to convert a converted file back to the original).

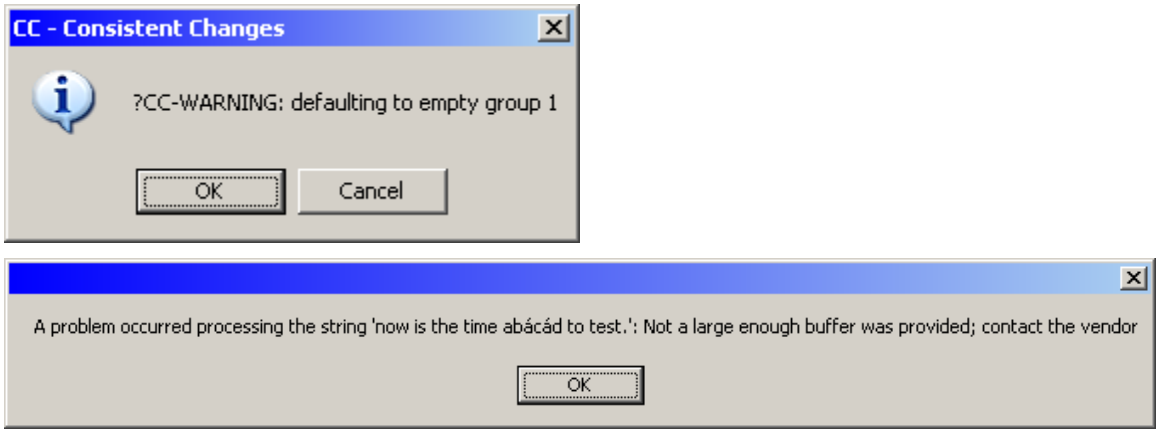
A CC table can use Unicode code points on both the match *and* replacement side of a change. A Unicode code point is specified with the u1254f syntax, where the value following “u” is the 1–6 digit hex value of the desired code point. CC will automatically convert this to the equivalent UTF-8 bytes and use them in the match or replacement. For encoding conversion, the match side will normally be standard character strings or decimal or hex values, while the replacement will normally be Unicode values. Here is an example of a typical command using a one-to-one mapping:

x91 > u2018 c LEFT SINGLE QUOTATION MARK

Note: Any non-ASCII code (above 0x7F) *must* be changed by the CC table or the resulting file will have illegal UTF-8 characters! Any UTF-8 character with the upper bit set *must* be part of a multibyte character sequence.

If you need to use movement commands with Unicode data, you have several options. Under normal operation *fwd, back, omit, prec, fol, any*, and *word* commands all work on bytes, which could easily mess up UTF-8 code points. These commands will work with UTF-8 code points by using the *utf8* command in the *begin* statement. You can also use UTF-8 specific commands such as *fwdu, backu, omitu, precu, folu, anyu*, and *wordu* regardless of the overall mode of the CC table.

Problems: If you see this dialog, followed by the second dialog when you click Cancel, it's probably because you have an older copy of cc32.dll in c:\Windows\System32. Even though SIL Encoding Converters installs a current copy of cc32.dll in c:\Program Files\Common Files\SIL, if some other program puts an older version in c:\Windows\System32, the program uses this older version. The solution is to copy the latest version of cc32.dll into c:\Windows\System32.



CP1252 to Unicode.cc is a sample CC table that will convert the standard Windows code page 1252 to Unicode. The ASCII part of this table (under x80) could actually be omitted since ASCII characters are unchanged in UTF-8. You could use this file as a starting point when building a converter for your data, only changing the output for your "hacked" characters.

4 TECkit

TECkit is an SIL program written and supported by Jonathan Kew that is specifically designed for encoding conversion. It provides good context sensitivity capabilities, where this is needed. It has various ways of defining and working with ranges of characters. You can usually write a single TECkit table to allow conversion to and from Unicode.

The source changes for TECkit are defined in a file with a .map extension. Compile this file to a .tec format to use during actual conversion. Whenever you make a compiled version, also provide the source table so people can see what it does and can modify it as needed. When SIL Encoding Converters uses a .map file as input, it automatically creates a compiled .tec file in the same directory.

Numerous TECkit tables are available from the NRSI Web site at http://scripts.sil.org/cms/scripts/page.php?site_id=nrsi&cat_id=ConversionMaps to convert legacy fonts or to convert standard data from a number of SIL branches.

The FieldWorks installation includes a TECkit map to convert SIL IPA93 encoded text to Unicode in the c:\Program Files\SIL\FieldWorks\Fonts\IPA93Mapping directory. You need to set up a converter for this map (silipa93.tec) before using it.

A TECkit file may contain one or more passes. Each pass begins with a header line that specifies the type of data expected in this pass.

```
pass(unicode)
```

The type inside the parentheses can be Byte, Unicode, Byte_Unicode, or Unicode_Byte. Note that in FieldWorks Bulk Edit, only TECkit files with type Unicode will be available for use in the Process tab. This is because all data in FieldWorks is Unicode, so Byte converters do not make sense in this context. From the Writing System Properties dialog or one of the Import dialogs, you can use a TECkit map with Unicode, Byte, or Byte_Unicode types as long as the output data is valid Unicode since input data can be byte or unicode. The pass type occurring in the file overrides the Conversion Type selection in the FieldWorks converter setup dialog. To see the real type in the TECkit file, go to the Encoding Converters dialog from the Converters tab of WritingSystemProperties, then in the Advanced tab, check the Type.

CP1252 to Unicode.map is a sample TECkit map that converts the standard Windows code page 1252 to and from Unicode. You can use this file as a starting point to build a converter for your data, and only change the output for your "hacked" characters. Unlike CC, you *must* include every code point in a TECkit table. Any character in your input file *not* included in the TECkit table is converted to U+FFFD REPLACEMENT CHARACTER. Here is a typical line from a TECkit map using one-to-one mapping:

```
0x91 <> U+2018 ; LEFT SINGLE QUOTATION MARK
```

The input and output side of each command can contain more than one value.

Windows-1252.map is another map that converts Windows code page 1252 to and from Unicode. It uses classes to shorten the table and also uses Unicode names for the output instead of hex values.

The SIL Converters package provides a TECkit Map Unicode Editor program that is quite useful for building TECkit maps. It provides the basic table in an editor and allows you to click in a second window to insert things such as character values and names while clicking on a font layout. It then shows input and output characters going both directions through the table.

Read the TECkit documentation for more information.

5 Windows code pages

SIL Encoding Converters provides access to all Windows code page converters that are installed on your system. These will handle most standard (nonhacked) fonts for any language.

Some useful examples

- Western European (Windows) [1252] converts characters using most standard Windows fonts (such as Times New Roman and Arial) to and from Unicode.
- Korean (EUC) [51949] converts Korean on most Far Eastern operating systems to and from Unicode. Use the Batang font to see the results in Unicode.
- Chinese Simplified (GB2312) [936] converts Chinese on most Far Eastern operating systems to and from Unicode. Use the Sim Sun font to see the results in Unicode.

If you have standard text on a non-US machine, it usually requires some experimentation to find a Windows converter that will convert the encoding to Unicode. You can run the text through the converter using different code pages until you find one that converts it properly.

6 ICU

ICU also provides many encoding converters but the FieldWorks installation eliminates those covered by Windows code pages to reduce the space used on the hard drive and the size of installers.

ICU also supplies a number of Unicode to Unicode transducers. There are quite a few that go from common scripts to a Latin transliterated form, or vice versa, such as Greek to Latin, and Latin to Greek. The following transducers can also be very helpful in exploring Unicode.

- Any to NFD: Converts data to NFD (decomposed normalization)
- Any to NFC: Converts data to NFC (composed normalization)
- Any to Hex Escape/Unicode: Converts data to U+hhhh Unicode hex values.
- Hex Escape to Any/Unicode: Converts Unicode U+hhhh hex values to actual code points.

These transducers can be used in various ways:

- in Microsoft Word using the Word macro package Data Conversion Macro dddd.dot
- in any application using ClipboardEC.exe for cut/paste operations,
- in SFM files using SFMConv.exe
- in XML files using SilConvertersXML.exe, and
- in Flex Bulk Edit operations.

7 Perl

SIL Encoding Converters supports Perl scripts written for Active State Perl or Perl Expressions 5.10.0. You can download these free of charge from http://www.activestate.com/Perl.plex or http://www.pxperl.com/?pxperl. One of these must be installed before you can use Perl scripts in SIL Encoding Converters. The following Perl script will convert the input string to lowercase:

```
$strOut = lc($strIn);
```

For more details on Perl converters, see the help information in Start...Programs…SIL Converters…Help…Help for Perl Expression Plug-in.

If you installed Perl after installing SIL Encoding Converters, you may need to manually register c:\Program Files\Common Files\SIL\3.0.0.0\PerlEC5100.dll before it will become available.

8 Python

SIL Encoding Converters supports Python programs written for Python.org or Active Python 2.5.4.4 (currently not newer versions). You can download these free of charge from www.codeplex.com/Wiki/View.aspx?ProjectName=IronPython, http://www.python.org/download/ or http://www.activestate.com/Products/ActivePython. One of these must be installed before you can use Python scripts in SIL Encoding Converters. The following Python function will convert the input string to uppercase.

```
def ToUpper(s):
    return unicode.upper(s)
```

IronPython 1.0.60816 works with SIL Encoding Converters if you create the following registry key: HKEY_LOCAL_MACHINE\SOFTWARE\Python\PythonCore\2.5 and add an InstallPath string variable that gives the path to your ipy.exe.

For more details on Python converters, see the help information in Start...Programs…SIL Converters…Help…Help for Python Script Plug-in.

If you installed Python after installing SIL Encoding Converters, you may need to manually register c:\Program Files\Common Files\SIL\2.6.1.0\PythonEC25.dll before it will become available.

9 Other converters

SIL Encoding Converters also provides several other converter engines that you may find useful. For more details on these converters, see Start…Programs…SIL Converters…Help.

- You can create a Compound Converter that chains multiple converters together, with the output of one going into the input of the next. For example, if you have a CC table that only works with NFC data, instead of modifying the table, you could use a Compound Converter that first uses an ‘Any to NFC’ ‘ICU Unicode to Unicode transducer’ before calling your CC table.
- A Primary-Fallback Compound Converter allows changes to be made by a primary converter, but if nothing changes, it then falls back to use a secondary converter.
- An AdaptIt Knowledge Base Lookup Converter allows word-for-word lookup and replacement using an AdaptIt knowledge base.

10 Clipboard Converter

You may have legacy data in Shoebox, Word, or some other program and want to paste some of this into a FieldWorks application. How can you convert the legacy text to Unicode? The Clipboard Converter that comes with the SIL Encoding Converter package provides a nice solution for this. With this program, you can select an encoding converter to be applied to data on the clipboard before it is pasted into your application.

Start the Clipboard EncConverter from the Windows Start…Programs…SIL Converters menu and it appears as an icon in your system tray. If you copy some text to the clipboard, right-click this icon and select the converter you want to apply. When you paste the text, it applies the selected converter to the pasted text. After a paste, the clipboard operation returns to normal. If you need to apply the same converter repeatedly, you need to select the converter *each time* prior to each paste.

You can also use the Clipboard Converter to add or remove converters from the repository. You can add a new converter by choosing Add Converter from the Right-click menu. You can choose Edit or Delete Converter from the right-click menu to modify or delete a converter. From the dialog, right-click the desired converter to edit or delete it. This allows you to add Perl, Python, or compound converters that cannot be added directly from within FieldWorks. Once a converter is added, FieldWorks can use the converter.

11 Word data conversion macro

The SIL Encoding Converter package provides several Microsoft Word macros that use these converters and transducers within Word. To activate one of the macros in Word, use the Tools…Templates and Add-Ins option, then click Add and select the .dot file you want to add.

Data Conversion Macro 0300.dot is a general converter that can run converters within Word using various options, such as limiting the changes to specific backslash codes for an SFM file. You can also add or remove converters from the repository from within this macro. Once activated, this macro can be accessed from the Tools…Data Conversion menu.

The installer also installs SpellFixer.dot and Consistent Spelling Checker 152sc.dot that can also be activated and used within Word.

When installed, these files are in the %USERPROFILE%\Application Data\Microsoft\Templates where %USERPROFILE% is C:\Documents and Settings* on Windows XP and c:\Users* on Vista and * is your Windows logon name.

12 Other SIL Converter programs

The SIL Encoding Converters package also installs some other programs that work with SIL Encoding converters.

- SFMConv.exe allows you to select a converter for each backslash code and then write out the converted SFM file.
- SilConvertersXML.exe allows you to do selective conversion in XML files.
- TECkit Map Unicode Editor program was discussed under the TECkit section.