



THE
POWER
TO KNOW.

Technical Paper

SAS 9.1.3 Service Pack 4 in a Unicode Environment

Table of Contents

Introduction	1
Concepts	2
Set-up a new Unicode session	3
Set-up a Unicode session based on an existing SAS session	4
Set-up a new Unicode server on the SAS Intelligence Platform	4
Configure a Unicode platform based on an existing SAS Intelligence Platform	6
Configuration checklist	7
SAS/ACCESS® in a Unicode environment	8
SAS/ACCESS interface to Oracle	8
Scenarios and how to configure	8
Configuration checklist	9
Current Status	9
Data types	9
Current limitations and workarounds	12
Workaround	13
Data sorting problem and workaround	13
Workaround	14
SAS/ACCESS® Interface to MySQL	14
Scenarios and how to configure	14
Current status	14
Current limitations and workarounds	15
SAS/ACCESS for Informix	16
Scenarios and How to Configure	16
Scenarios	16
Configuration	16
Current status	16
Access for external files	17
Scenarios	17
Current status	17
Current limitations and workarounds	17
Base SAS, Scalable Performance Database Engine in a Unicode Environment...20	
The Unicode SAS session	20
Current status	20
Current limitations and possible workarounds	20
Scalable Performance Data Engine in a Unicode SAS session	21

Current Status	21
Current limitations and possible workarounds	21
Scalable Performance Data Server	22
SAS servers in a Unicode environment.....	23
SAS clients in a Unicode environment	24
Current Status	24
Current limitations and workarounds	24
Font Display of Java Client Applications.....	24
DIS Limitations	25
AMO Limitation.....	25
SAS® Enterprise Guide® limitation	26
Additional font installation	27
SAS/CONNECT® in a Unicode environment.....	27

Introduction

The requirement of accessing, storing and analyzing multilingual data using SAS[®] software is rapidly increasing worldwide. SAS 9.1.3 provides the capabilities to meet this business requirement with Unicode support.

Unicode is an industry standard allowing computers to consistently represent and manipulate the text contained in the writing systems around the world. This paper describes how to configure a Unicode server on the SAS Intelligence Platform to process multilingual data. Potential issues and limitations with SAS 9.1.3 in a Unicode environment are also discussed as well as possible workarounds for dealing with these issues.

This paper outlines

- How to configure and start a SAS[®] Unicode server.
- The current status of accessing multilingual data from existing data sources and legacy systems, including accessing data from existing SAS systems, database management systems (DBMS) and external files.
- The current status of managing and exploring multilingual data and developing applications with the Unicode SAS workspace server and stored process server.
- The current status of managing and exploring multilingual data with SAS clients in a Unicode environment, including Windows clients, Java clients and Web clients.

Concepts

This document assumes that the reader has an understanding of the following concepts:

- Character set.
- Encoding.
- Transcoding.
- Unicode.
- Legacy encoding.
- SAS with DBCS extensions.
- Working with a SAS Unicode server.

A character set is a collection of symbols and punctuation marks used in a single language or in a group of languages.

An encoding is the association of a unique numeric value with each symbol and punctuation mark in a character set. There are two groups or types of encodings: single byte character set (SBCS) encodings and DBCS encodings. SBCS encodings represent each character in a single byte. DBCS encodings require a varying number of bytes to represent each character. A more appropriate term for “DBCS” is multi-byte character set (MBCS). MBCS is sometimes used as a synonym for DBCS.

SBCS encodings are limited to 256 possible characters. DBCS encodings can represent more than 256 characters. Beginning with SAS 8.2, each SAS session has one session encoding. The session encoding is set using the LOCALE or ENCODING option. Transcoding is the process of converting from one encoding to another.

Unicode is the universal character encoding that supports the interchange, processing, and display of characters and symbols found in the world’s writing systems. Other character encodings are limited to subsets of all languages. Often the subsets are limited to a region (for example: Windows Latin 1 [WLATIN1] represents the characters of the US and Western Europe on Windows). UTF8 is the Unicode encoding form where characters are represented in sequence of one to four bytes in length. A legacy encoding is one of the DBCS or SBCS encodings which pre-date the Unicode standards. Legacy encodings are limited to the characters from a single language or a group of languages.

SAS DBCS extensions are an optional supplement to Base SAS[®] that provide support for DBCS encodings. In SAS[®]9 the DBCS extensions are available on the SAS software media. When you install SAS software, you can choose to install with or without the DBCS extensions. SAS 9 uses the DBCS extensions to support the UTF8 encoding as a SAS session encoding. In this paper, I will refer

to the SAS DBCS system running with a session encoding of UTF8 as the Unicode server.¹

Configuring SAS 9.1.3 Service Pack 4 with a Unicode server

This section describes how to configure a Unicode environment, including a Unicode session and a BI Platform.

Set-up a new Unicode session

A Unicode session is a SAS session with UTF-8 encoding. During SAS installation, make sure that the 'English with DBCS and Unicode Support' language option is selected.

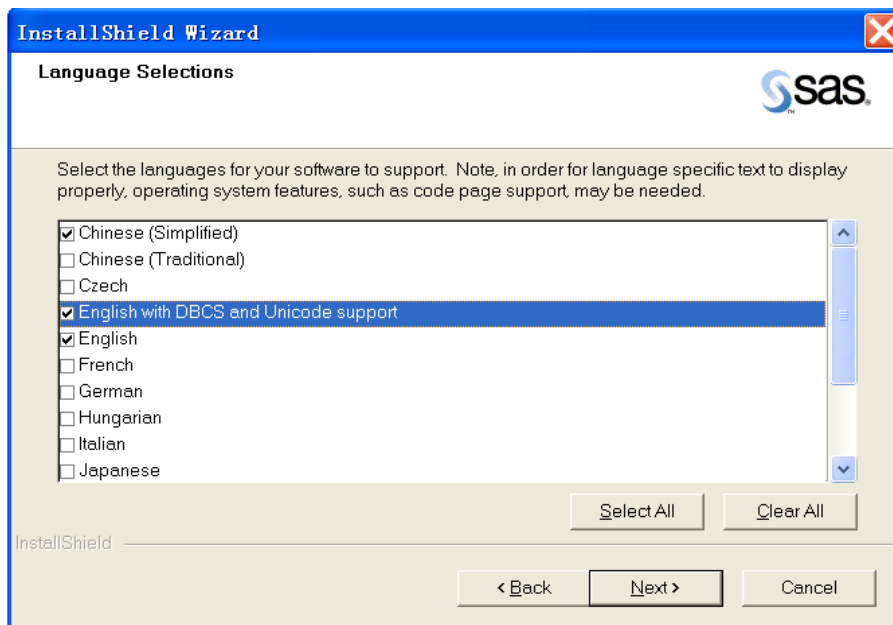


Figure 1: InstallShield Wizard

Modify SASV9.CFG file in directory `!SASROOT\SAS 9.1\nls\en`

Add `-ENCODING UTF-8`

When you invoke SAS you will be running the English version of the SAS System with Unicode support.

¹ Cf. Stephen Beatrous. 2003. "Multilingual Computing with the 9.1 SAS Unicode Server." SAS Presents, Twenty-Eighth Annual SAS Users Group International Conference, Seattle, WA.

Set-up a Unicode session based on an existing SAS session

A Unicode session can also be set-up based on an existing English version of the SAS System. For the time being, there are issues to build a proper SAS Unicode session based on non-English locales and or languages with SAS 9.1.3 Service Pack 4. .

1. Modify SASV9.CFG file in directory `!SASROOT\SAS 9.1\nls\en`
2. Add '-ENCODING UTF-8'
3. Add "!sasext0\dbc\sbcs\sasexe" to -PATH section:

```
/* Setup the SAS System load image search paths definition
*/-PATH
    ("!sasext0\dbc\sbcs\sasexe"
    !sasroot\core\sasexe"
    ... ..
```

Note: The SAS DBCS extensions must have been installed prior to making this modification. Otherwise, the 'dbc' directory will not exist in your operating system.

Set- up a new Unicode server on the SAS Intelligence Platform

A Unicode server must be setup on a SAS Intelligence Platform to make it possible for SBCS and DBCS data to be processed at the same time During SAS installation, 'English with DBCS and Unicode support' should be selected as the SAS System default.

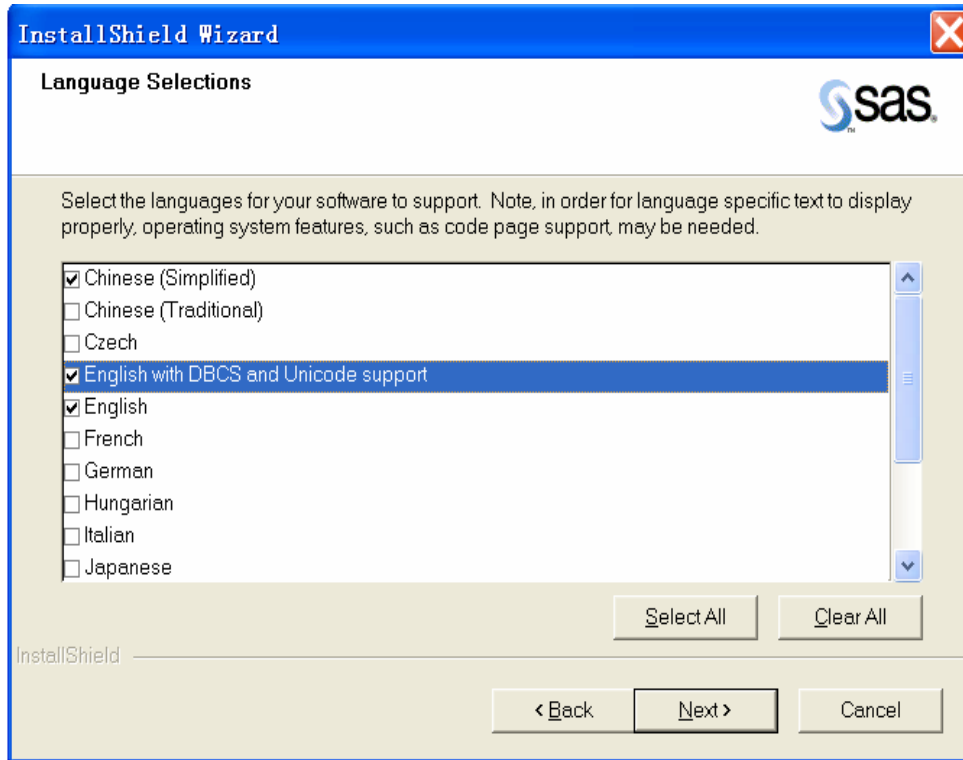


Figure 2: Language selection window

While running the SAS Configuration Wizard, select 'English' in language selection dialog box

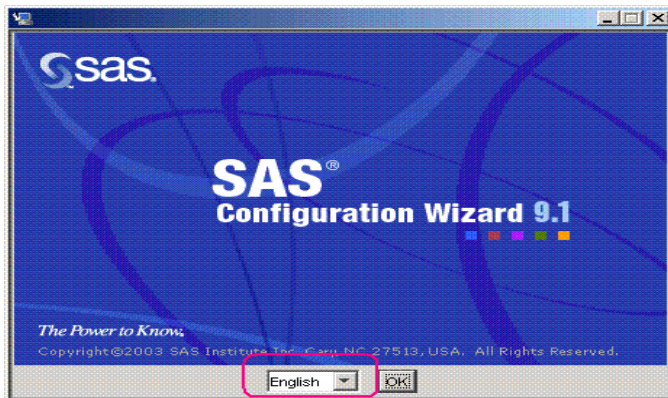


Figure 3: SAS Configuration Wizard

1. Modify 'SASV9.CFG' file in directory '!SASROOT\SAS 9.1\nls\ld'

Add '-ENCODING UTF-8'

2. Restart SAS Servers

After modifying 'SASV9.CFG', SAS servers must be restarted. To restart SAS servers:

Select Start -> Setting -> Control panel -> Administrative Tools -> Service. Right click on 'SAS Lev1 MS - 9.1' and then select 'Restart'.

3. All other SAS servers must be restarted the same way.

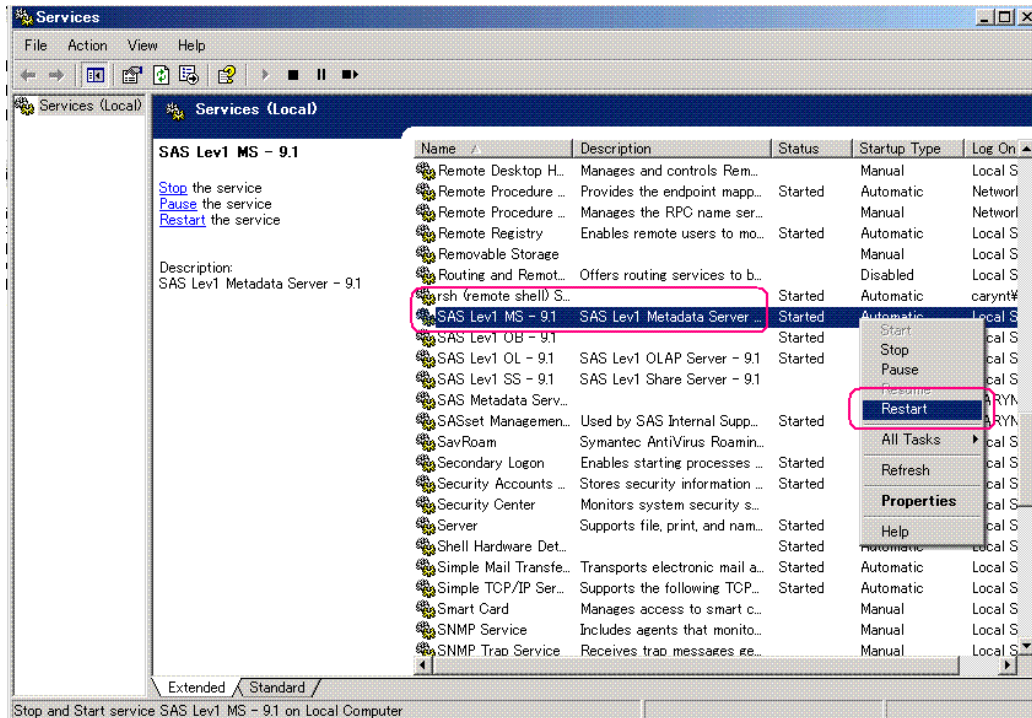


Figure 4: Stop and start service window

Configure a Unicode BI platform based on an existing SAS Intelligence Platform

The following steps can be taken to change an existing SAS Intelligence Platform to a Unicode BI Platform.

1. Modify the 'SASV9.CFG' file of the default SAS System. For example: if Simplified Chinese is selected as the default SAS System, the 'SASV9.CFG' file should be under '!SASROOT\SAS 9.1\nls\zh' Add '-ENCODING UTF-8'

2. If the language of the default SAS System is a SBCS languages, then add "!sasext0\dbcs\sasexe" to -PATH section:

```
/* Set-up the SAS System load image search paths definition */
-PATH (
    "!sasext0\dbcs\sasexe"
    "!sasroot\core\sasexe"
```

Note: The SAS DBCS extensions must have been installed prior to making this modification. Otherwise, the 'dbcs' directory will not exist in your operating system.

When the language selected within the SAS Configuration Wizard is a DBCS language:

1. Modify 'startShareServer.sas' which is under SAS deployment directory, such as 'C:\SAS\Unicode\Levl\SASMain\ShareServer'.
 - a. Replace the DBCS Language word for "Server" with the English word 'Server'.
 - b. After the modification, it should look like:

```
%let SVRNAME=SASMain - SAS/SHARE Server
```
2. The same change must be made in SMC. It should be: 'SASMain - SAS/SHARE Server' under 'Server Manager' of 'Environment Management'.
3. Restart SAS Servers
4. Restart SAS servers as described in the section Set-up a new Unicode server on an existing SAS Intelligence Platform.

A note about the SAS Metadata Server: The default encoding does not affect the metadata server's handling of encoding for data sets. The metadata server maintains the encoding in which the data set was stored. You should be able to change the default encoding without disturbing the metadata server.

Configuration checklist

1. Has the SAS DBCS component been installed? To check this, confirm that the folder: "!SASRoot\dbcs\sasexe" exists.
2. Has the 'SASV9.CFG' configuration file been changed correctly? Check whether '-ENCODING UTF-8' is added; Check that the DBCS component has been enabled by adding "!sasext0\dbcs\sasexe" to PATH section:

```
-PATH (
    "!sasext0\dbcs\sasexe"
    "!sasroot\core\sasexe"
    ... ..
```

3. Has SAS been invoked with the UTF-8 encoding? To confirm the encoding, submit the following code in a SAS session: `Proc options option=encoding; run;`

SAS/ACCESS[®] in a Unicode environment

SAS/ACCESS interface to Oracle

Scenarios and how to configure

Scenarios

Oracle databases support Unicode. It can process multi-language data correctly if its character sets are set properly. Unicode Oracle data are accessed by SAS/ACCESS interface to Oracle engine.

Configuration

1. Access configuration through a SAS Management Console.
2. Add a new user group for Oracle database authentication from the user manager.
3. Create a new server for Oracle database from the server manager.
4. Create a new database schema from the data library manager.
5. Define a new library for Oracle data.

SAS/ACCESS configuration via the library statement in the SAS Session

```
LIBNAME ***** ORACLE PATH=***** SCHEMA=***** USER=*****  
PASSWORD=***** ;
```

PATH should be the Oracle local net service name. SCHEMA should be the same as the schema name in Oracle. For the exact value of the local net service name, schema, user, and password please contact the Oracle administrator.

Setting the environment variable

To correctly access Oracle data, the `NLS_LANG` environment variable must be added to the operating system where the SAS/ACCESS interface to Oracle is running.

```
NLS_LANG=LANGUAGE_TERRITORY.CHARACTERSET
```

Values for LANGUAGE and TERRITORY should be set according to the local operating system setting. CHARACTERSET tells Oracle the encoding of data being sent and received. When the SAS session is UTF-8 encoded, CHARACTERSET should be set to AL32UTF8 or UTF8 (for Oracle 8i, only UTF8 is effective).

For example: if SAS/ACCESS interface to Oracle is running on an English operating system, the NLS_LANG environment variable should be set to:

```
NLS_LANG=AMERICAN_AMERICA.AL32UTF8
```

Configuration checklist

To determine the character set and national character set of an Oracle database submit the following code in Oracle.

```
SELECT PARAMETER, VALUE
FROM NLS_DATABASE_PARAMETERS
WHERE PARAMETER IN ( 'NLS_CHARACTERSET', 'NLS_NCHAR_CHARACTERSET' );
```

Ensure that the NLS_LANG environment variable has been correctly set on the OS where SAS/ACCESS Interface to Oracle is running as described in previous section.

For example:

```
NLS_LANG=AMERICAN_AMERICA.AL32UTF8
```

Current Status

Data types²

The following table shows the default SAS variable formats that SAS/ACCESS assigns to Oracle data types during input operations when you use the LIBNAME statement.

LIBNAME Statement: Default SAS formats for Oracle data types

² "SAS/ACCESS Software," SAS OnlineDoc, Version 9.1. CD-ROM. (Cary, NC: SAS Institute Inc., 2003).

Oracle Data Type	Default SAS Format
CHAR(n)	\$n.
VARCHAR2(n)	\$n.
NUMBER	none (BEST. on OS/390)
NUMBER(p)	w.(BEST. on OS/390)
NUMBER(p, s)	w.d
DATE	DATETIME20.
LONG	\$1024.
RAW(n)	\$HEXw.
LONG RAW	\$HEX2048.

Oracle data types that are omitted from this table are not supported by SAS/ACCESS. When Oracle data falls outside valid SAS data ranges, the values are usually counted as missing.

The following table shows the default Oracle data types that SAS/ACCESS assigns to SAS variable formats during output operations when you use the LIBNAME statement.

LIBNAME Statement: Default Oracle data types for SAS formats.

SAS Variable Format	Oracle Data Type
\$w.	VARCHAR2(n) ^{***}
w. with SAS format name of NULL	NUMBER(p)
w.d with SAS format name of NULL	NUMBER(p,s)
all other numerics *	NUMBER (NUMBER(38,10) on OS/390)
datetimew.d	DATE
datew.	DATE
time. **	DATE
<p>* Includes all SAS numeric formats, such as BINARY8 and E10.0.</p> <p>** Includes all SAS time formats, such as TODw,d and HHMMw,d.</p> <p>*** If the SAS char variable format is greater than \$4000, the default Oracle data type is LONG. If you use Oracle7 and the CHAR variable format is between \$2000. and \$4000. then use the DBTYPE= option to change the default VARCHAR2 type to LONG.</p>	

Encoding and column length in Oracle

In Oracle 9i, national character set only supports AL16UTF16 and UTF8. AL16UTF16 cannot be used as character set.³

By default, in Oracle 9i CHAR & VARCHAR2 are measured in bytes, also called byte semantics. For example, CHAR(6) means it can contain two DBCS characters if character set is AL32UTF8 / UTF8 and contain 3 DBCS characters if character set is ZHS16GBK. However the length of CHAR & VARCHAR2 can also be measured by character by using:

1. Expression such as "VARCHAR2(10 CHAR)"
2. Or set the value of NLS_LENGTH_SEMANTICS parameter to CHAR.

NCHAR, NVARCHAR2, and NCLOB data types have been redefined to support Unicode data only. You can store the data in either UTF-8 or UTF-16 encodings. It is controlled by the value of NATIONAL_CHARACTER_SET.

NCHAR, NVARCHAR2, CLOB, and NCLOB columns are always measured by character. For example: NCHAR(2) means it can contain 2 characters regardless of national character set.

How SAS/ACCESS interface to Oracle interprets length in Oracle

When reading Oracle data into a SAS session:

- SAS does not change the length for CHAR & VARCHAR2 variables if byte semantics.
- SAS quadruples the length for CHAR & VARCHAR2 variables if character semantics and Oracle character set is AL32UTF8; SAS doubles the length for CHAR & VARCHAR2 variables if character semantics and Oracle character set is ZHS16GBK.
- SAS doubles the length for NCHAR & NVARCHAR2 variables if national character set is AL16UTF16.
- SAS triples the length for NCHAR & NVARCHAR2 variables if national character set is UTF8.

Current limitations and workarounds

The limitations listed below are based on the assumption that the length semantics in Oracle is byte Semantics, which is default for Oracle.

³ Oracle Corporation, Choosing a Character Set, Oracle9i Database Globalization Support Guide Release 2 (9.2), 2002. Available http://download-east.oracle.com/docs/cd/B10501_01/server.920/a96529/ch2.htm#100806.

Variable length problem and workaround

We will encounter limitations in following cases:

Character Set	National Character Set	
	AL16UTF16	UTF8
AL32UTF8 / UTF8	NCHAR & NVARCHAR2 have length problem	OK
ZHS16GBK	CHAR & VARCHAR2, NCHAR & NVARCHAR2 all have length problem	CHAR & VARCHAR2 have length problem

When AL16UTF16 is selected as NATIONAL CHARACTER SET, it will cause a variable length problem because SAS only doubles the length of Oracle column. Data truncation will happen on NCHAR and NVARCHAR2 data types because data are measured in bytes in SAS. With the UTF-8 encoding, one DBCS character will take three bytes. Although the length of the NCHAR and NVARCHAR2 Oracle columns is doubled by SAS/ACCESS by default, it is still not long enough which can sometimes result in data truncation. This issue will always occur when the National Character Set is AL16UTF16, regardless of the character set value set.

Workaround

There is no workaround to this issue for SAS 9.1.3 Service Pack 4

In SAS 9.2, this issue is resolved by four options:

1. ADJUST_BYTE_SEMANTIC_COLUMN_LENGTHS
2. ADJUST_NCHAR_COLUMN_LENGTHS
3. DBCLIENT_MAX_BYTES
4. DBSERVER_MAX_BYTES

Data sorting problem and workaround

When merging data from Oracle and SAS data sets, if the BY variable for both contains NULL an error will occur.

This error occurs because NULL is always treated as the biggest value in Oracle and the smallest value in SAS. Data must be correctly pre-processed before merging to avoid this error.

Workaround: First import the table into SAS from Oracle, perform the sort in SAS and then merge the data after sorting.

SAS/ACCESS[®] Interface to MySQL

Scenarios and how to configure

Scenarios

The MySQL Server data is accessed by the SAS/ACCESS Interface to ODBC.

Configuration

Define ODBC Data Source

An ODBC data source should be defined on the operating system where SAS/ACCESS Interface to ODBC is running.

A data source can be created by selecting an ODBC driver from the ODBC Administrator in the Windows Control Panel. Information, for example, data source name, user ID, password, description, and server name, is required by the driver to make a connection to the desired data. During operation, a client application usually requests a connection to a named data source, not just to a specific ODBC driver.

An ODBC Administrator tool is not available in a UNIX environment such as HP-UX, AIX or Solaris. During an install, the driver creates a generic .odbc.ini file that can be edited to define your own data sources.⁴

Library Definition in SAS Session

```
LIBNAME ***** ODBC DATASRC=***** ;
```

Here, DATASRC should be defined as a MySQL Server data source name in ODBC.

Current status

Unicode supports data types and encoding. The Unicode data types listed below are supported in MySQL Server, other data types cannot be stored as Unicode:

- NCHAR
- NVARCHAR / NVARCHAR(MAX)
- NTEXT

⁴ SAS Institute Inc., SAS® 9.1.3 Intelligence Platform: System Administration Guide. (Cary, NC: SAS Institute Inc., 2006).

The Unicode encoding in MySQL Server is UTF-16.

Variable length

When reading MySQL Server data into a UTF-8 SAS session via ODBC,

- SAS doubles the length for NCHAR & NVARCHAR variables.
- Format and informat for original DATE column is DATETIME22.3.

Current limitations and workarounds

Transcoding problem

Currently, SAS/ACCESS Interface to ODBC cannot support Unicode. DBCS data and single-byte non-ASCII characters cannot be correctly processed.

In a SAS UTF-8 session, data imported appear as question marks. Although the encoding property of imported data is UTF-8, the real encoding of the data is not changed to UTF-8. There is a workaround to solve this problem, which might not be practical

Workaround: For SBCS data, like Spanish: In a SAS UTF-8 session, after reading data from MS SQL Server to a SAS data set, the data set's encoding must be modified to its original encoding. The data set has to be transcoded to UTF-8.

A code example for Spanish data is:

```
/* Load data into SAS data set from MS SQL Server
   data stagings.custtype_w1;
   set sqldata.Cust_Type_es;
/* sqldata is a library defined in SQL server via SAS/Access*/
   run;
/* Set the correct encoding to 'WLATIN1' */
   proc datasets lib=stagings;
   modify custtype_w1/correctencoding=wlatin1;
   run;
/* Transcode data from 'Wlatin1' to 'UTF-8' */
   data stagings.custtype_u8;
   set stagings.custtype_w1;
   run;
```

- For DBCS data, like Chinese

The Language for programs that are not Unicode on Windows must be changed from 'English (United States)' to the language it belongs to, such as Chinese (PRC) for Chinese.

The operating system has to be restarted. In a SAS UTF-8 session, after reading the data from

MySQL Server to a SAS data set, the data set's encoding must be modified to its real encoding, such as EUC-CN for Chinese. Then the data set must be transcoded to UTF-8.

SAS/ACCESS for Informix

Scenarios and How to Configure

Scenarios

Informix data is accessed by SAS/ACCESS Interface to ODBC.

Configuration

Define an ODBC data source

An ODBC data source should be defined on the operating system where SAS/ACCESS Interface to ODBC is running.

A data source can be created by selecting an ODBC driver from the ODBC Administrator in the Windows Control Panel. Information (for example, data source name, user ID, password, description, and server name) is required by the driver to make a connection to the desired data. During operation, a client application usually requests a connection to a named data source, not just to a specific ODBC driver.

An ODBC Administrator tool is not available in a UNIX environment such as HP-UX, AIX, or Solaris. During an install, the driver creates a generic .odbc.ini file that can be edited to define your own data sources.^[3]

Library Definition in SAS Session

```
LIBNAME ***** ODBC DATASRC=***** SCHEMA=*****;
```

DATASRC should be defined as the Informix data source named in ODBC. SCHEMA is the Informix database name.

Current status

According to our investigation, Informix ODBC driver does not do the transcoding work at all. When data is imported from Informix to a UTF8 SAS session, the real encoding of the data remains the same as the original encoding in Informix even though the encoding property of the SAS data set is set the same as the SAS session.

To access Informix data correctly, you must:

- Know the original data's correct encoding in Informix before loading the data into a SAS data set;
- After loading the data into a SAS data set, manually change the value of the encoding property of the SAS dataset to the Informix encoding value;
- When necessary, transcode the dataset to a UTF8 encoding.

Access for external files

Scenarios

External files can be accessed by SAS directly or through a PCFILES engine.

Current status

According to our investigation, a UTF-8 SAS session can support the following encoded flat files.

- UTF-8
- UTF-16 with some limitations. Refer to the section Current limitations and workarounds for details.

For SAS 9.1.3:

- PC File Server does not support Unicode.
- Microsoft Excel files cannot be accessed by UTF-8 SAS session.

Refer to Current limitation and workaround for detailed information.

Current limitations and workarounds***UTF-16 Files***

If the UTF-16 file contains '1A'x, SAS will stop reading the source file when it encounters '1A'x character.

The IGNOREDOEOF option must be added to the INFILE statement because SAS treats '1A'x character as an end-of-file (EOF) character. SAS will stop reading the source file when it encounters such character. Adding the IGNOREDOEOF option to the INFILE statement will disable the internal EOF check. An example is shown below:

```
FILENAME tempfile '*' ENCODING="UTF-16";
DATA out;
    INFILE tempfile IGNOREDOEOF;
    INPUT ... ;
RUN;
PC File Server
```

In SAS 9.1.3, PC File Server does not support Unicode at all. DBCS data will be displayed as garbage when PC File Server is running on an English Windows operating system. When PC File Server is running on a DBCS language Windows environment, other DBCS languages' data will be displayed as garbage and the data of the local language will be truncated.

In SAS 9.2, a new LIBNAME option named UNICODE_CONNECTION has been created to resolve this problem. When this option is set to 'YES', PC File Server can support Unicode and the data truncation problem is resolved as well.

For example:

```
LIBNAME temp PCFILES SERVER="crdvm524.apac.sas.com" PATH="D:\temp\temp.xls"
UNICODE_CONNECTION=YES;
Excel File
```

In SAS 9.1.3, an Excel file can only be correctly accessed by SAS when the SAS session encoding is consistent with the Excel file's character set. For example, when reading an Excel spreadsheet with Japanese character set by SHIFT-JIS SAS session, it works fine. However, an Excel file that contains multi-language data cannot be correctly accessed by a UTF-8 SAS session. Garbage characters will be displayed.

Note: In our testing, the Excel file was only accessed through LIBNAME and PROC IMPORT.

There is a workaround for this problem.

1. In Microsoft Excel on Chinese Windows, save the Excel file as Unicode text.

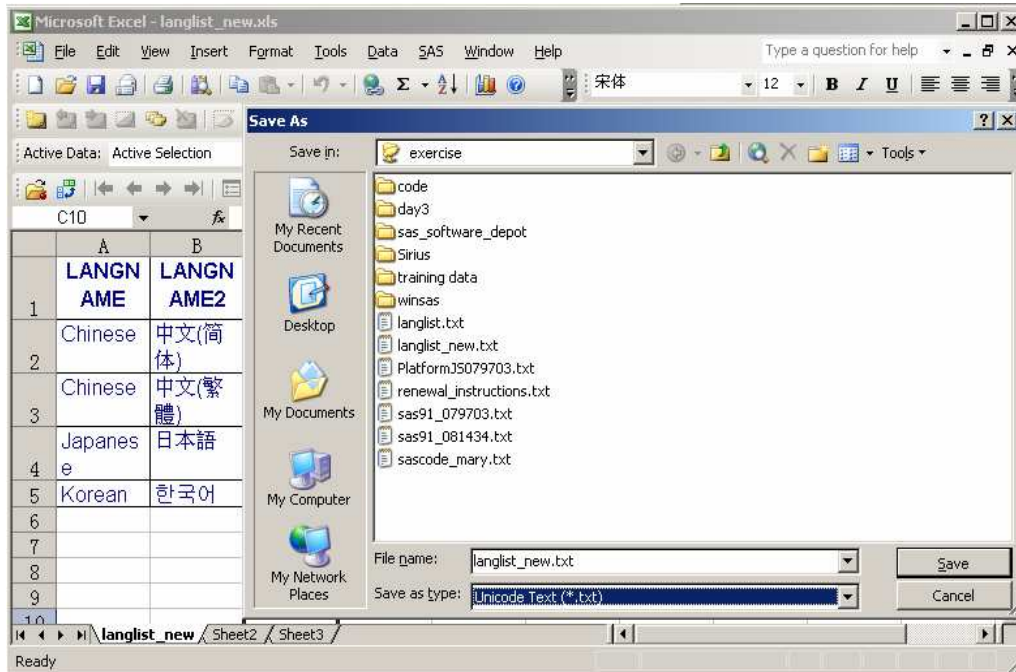


Figure 5 Import Unicode text files

2. In a UTF-8 SAS session, import the Unicode *.txt file into SAS by submitting a SAS program like the one shown below.

```
FILENAME IN "C:\langlist_new.txt" Encoding="utf-16";
PROC IMPORT OUT= WORK.langlist_new
            DATAFILE= IN
            DBMS=TAB REPLACE;
            GETNAMES=YES;
            DATAROW=2;
RUN;
```

Then, data can be correctly processed.

In SAS 9.2, this problem has been solved. A new LIBNAME and data set option – DBENCODING has been added, and its default value is 'UTF-16' for UTF-8 session.

Base SAS, Scalable Performance Data Engine in a Unicode Environment

The Unicode SAS session

Current status

Overall Base SAS with UTF8 encoding works well. Most of the features including CEDA work well.

Current limitations and possible workarounds

Program with Multilanguage characters

Programs with Unicode characters can be successfully displayed and submitted in SAS 9.1 (UTF-8) legacy program editor. However in the enhanced program editor, the multilingual characters cannot be correctly displayed.

DMS support

Usually we view a data set that contains multilingual characters via Viewtable in a SAS session. However, sometimes the result we get from Viewtable is misleading.

Usually data can be displayed in Viewtable. However if the data is highlighted, the text will be garbage characters.

In some cases, for example, when we view the data which is at DBMS tables via SAS/ACCESS Interface to ODBC, the data will be displayed as garbage characters in Viewtable, however if you highlight the data cell, you can see the right data text.

The rule is that if SAS can successfully recognize the encoding of the data, the data will be correctly displayed in the Viewtable unless you highlight the data cell. But sometimes, especially when SAS is working with a third party DBMS, SAS cannot correctly tell the encoding of the data.

The correct approach to view the multilingual data in a dataset is to make use of ODS HTML output.

Variable length

When transcoding data to UTF8 encoding, if the variable length in the target data set is not long

enough, the data step will stop, rather than truncate the data as previously thought. Usually we will get following errors.

```
ERROR: Some character data was lost during transcoding in the dataset
WORK.CUSTDIM.
      NOTE: The data step has been abnormally terminated.
      NOTE: The SAS System stopped processing this step because of errors.
... ..
```

This problem occurs because in the DATA step, the target variable's length remains the same as the source variable's length. However, the data length will become longer after transcoding making the target variable length not long enough to hold the transcoded data. As a result, the DATA step will stop, instead of truncating the data.

To solve this problem, the target variable's length needs to be set manually to ensure that it is long enough or use the CVP engine.

UTF-16 Encoding

Currently, we cannot get a UTF-16 encoded SAS session. The following error message will be received if you attempt to set a SAS data set's encoding to UTF-16: 'Invalid value for the ENCODING option.'

However SAS can read data from a UTF-16 encoded txt file.

Scalable Performance Data Engine in a Unicode SAS session

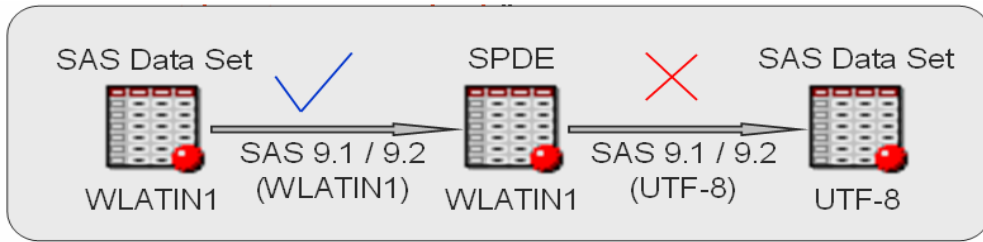
Current Status

Overall the Scalable Performance Data Engine in a UTF-8 SAS session works well. Most features, including indexing work well. By default, data under the Scalable Performance Data Engine are sorted by Unicode binary order for both SAS 9.1 and SAS 9.2.

There are some limitations as described below.

Current limitations and possible workarounds

SPDE engine cannot read SPDE data sets that are in an encoding that is different than the SAS session encoding. An error message will be received when doing this. "Error: Unable to open data file because its file encoding differs from the SAS session encoding and cannot be transcoded." The diagram below illustrates the situation.



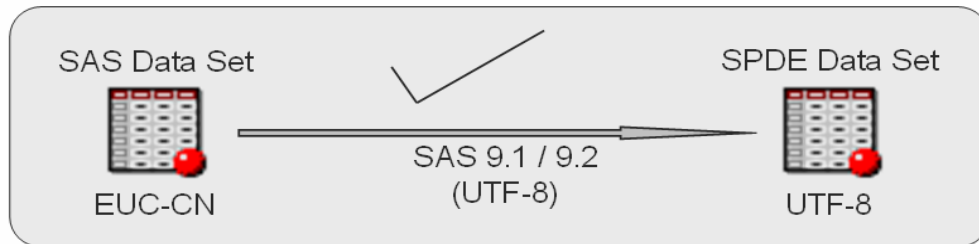
In addition, the ENCODING option is not implemented in SPDE engine.

In a SAS 9.1 / 9.2 UTF-8 Session, a warning will occur if you try to submit following code:

```
data spdetest.test(encoding="wlatin1");
    /* spdetest is a SPDE library */
    set spdstest.Custdim_es;
run;
```

"WARNING 76-63: The option ENCODING is not implemented in the SPDE engine."

However, data in a Base SAS dataset with an encoding that does not match the native SAS session can be imported to SPDE library. The diagram below illustrates the situation.



Scalable Performance Data Server

Scalable Performance Data Server 4.4 works well in a Unicode environment. SPDS 4.4 indexing works well under Unicode environment. By default, data is sorted by Unicode binary order.

SAS servers in a Unicode environment

Workspace Server and Stored Process Server support Unicode. Stored Process can be successfully executed without any errors both on the workspace server and stored process server. Cubes can be correctly based on UTF8 data in under UTF8 SAS environment. Information maps and reports for UTF8 cubes work well. Regarding data Collation, in reports based on UTF8 cubes, data is sorted by Unicode. This is also true for Excel and SAS Web Report Studio.

SAS clients in a Unicode environment

Current Status

SAS Web application clients such as WRS and IDP work well under Unicode environment.

SAS Java clients such as DI Studio, IMS and SMC work well with some limitations.

SAS Windows clients such as EG and AMO also work well, with some limitations.

Current limitations and workarounds

Font Display of Java Client Applications

SAS Java client applications, such as DI Studio, IMS and SMC, cannot support Unicode display by default. To support Unicode display, '...\JRE\...\lib\font.properties' used by each application must be modified and the languages' fonts need to be added to each section.

SAS users should adjust the priority of the fonts in the file according to their requirements. The filename definition for each language's font must also be added. An example is shown below:

```
# Component Font Mappings
#
dialog.0=Arial,ANSI_CHARSET
dialog.1=\u5b8b\u4f53,GB2312_CHARSET      /* Chinese Font */
dialog.2=\uff2d\uff33 \u30b4\u30b7\u30c3\u30af,SHIFTJIS_CHARSET /*
Japanese Font */
dialog.3=Arial Unicode MS,UNICODE_CHARSET      /* Unicode Font */
dialog.4=WingDings,SYMBOL_CHARSET
dialog.5=Symbol,SYMBOL_CHARSET

dialog.bold.0=Arial Bold,ANSI_CHARSET
dialog.bold.1=\u5b8b\u4f53,GB2312_CHARSET
dialog.bold.2=\uff2d\uff33 \u30b4\u30b7\u30c3\u30af,SHIFTJIS_CHARSET
dialog.bold.3=Arial Unicode MS,UNICODE_CHARSET
dialog.bold.4=WingDings,SYMBOL_CHARSET
dialog.bold.5=Symbol,SYMBOL_CHARSET

dialog.italic.0=Arial Italic,ANSI_CHARSET
```

```

dialog.italic.1=\u5b8b\u4f53,GB2312_CHARSET
dialog.italic.2=\uff2d\uff33 \u30b4\u30b7\u30c3\u30af,SHIFTJIS_CHARSET
dialog.italic.3=Arial Unicode MS,UNICODE_CHARSET
dialog.italic.4=WingDings,SYMBOL_CHARSET
dialog.italic.5=Symbol,SYMBOL_CHARSET
.....
# Font File Names
#
filename.\u5b8b\u4f53=SIMSUN.TTC
filename.\uff2d\uff33_\u30b4\u30b7\u30c3\u30af=MSGOTHIC.TTC
filename.Arial_Unicode_MS=ARIALUNI.TTF

```

DIS Limitations

Source Designer Wizard

When defining delimited external file as source data via the Source Designer Wizard in DI Studio, data will be truncated if column length is not long enough.

During the definition process, regardless of the method selected, column length cannot be automatically set correctly. If 'Auto Fill' is selected, column length will be defined as the longest column value's character length. So, longer DBCS strings will be truncated. If column name is imported from file, column length will be defined as the length of column name and may not long enough as well. To avoid this problem, column length definitions must be manually changed.

User Written Code

In DI Studio, there is a data transformation called "User Written Code." When a user written SAS program that contains multi-language characters is specified in this node, the job cannot run successfully. Code which includes multi-language characters will be ignored.

To solve this problem, the SAS program source must be saved as type of 'UTF-8 – NO BOM'. For example, if UltraEdit is used to edit the source of a SAS program, in addition to saving the SAS program as a type of 'UTF-8 – NO BOM', the 'Auto detect UTF-8 files' option in the UltraEdit configuration must be checked off as well. Otherwise, the BOM will exist in source code which will cause trouble.

AMO Limitation

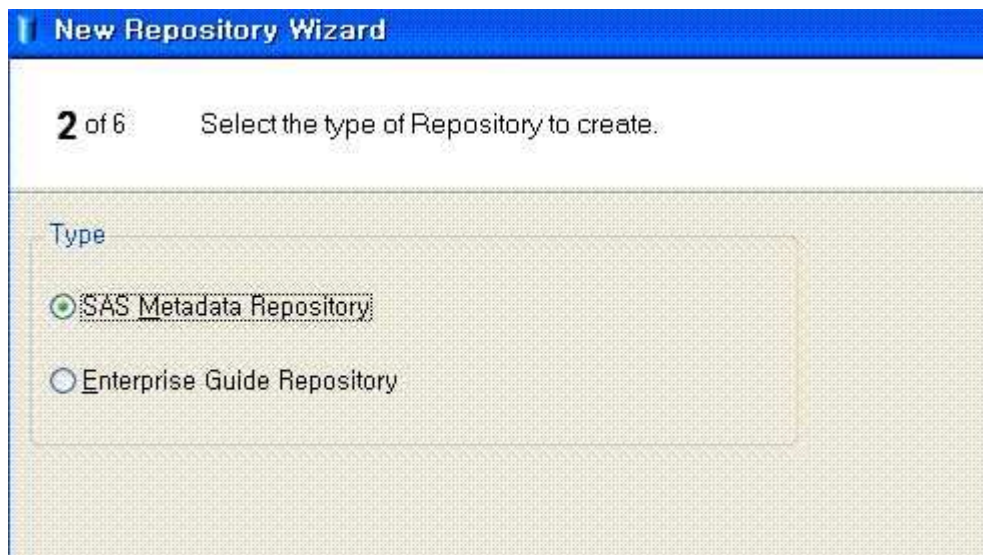
The log in AMO cannot support Unicode display and it has an operating system dependency.

SAS® Enterprise Guide® limitation

Garbage characters will display and transcoding errors will appear when importing a multi-language Excel file into EG. For example, errors occur when importing a Japanese Excel file in EG (Korea) pointing to SAS (Korea).

The following steps can be used to solve the problem described in the example above:

1. Replace the DBCS characters that comprise the name of the Logical Workspace Server with English characters via SMC.
2. Change the session encoding of SAS servers to UTF-8. Then, restart SAS servers.
3. In EG, create a 'SAS Metadata Repository' to connect to a UTF-8 Metadata Server and Workspace Server Please see below screen shot.



4. On the Windows OS where EG (Korea) runs, the system setting 'Language for non-Unicode programs' under 'Regional and Language Options' must be changed to Japanese. Then, restart the OS.

5. Japanese Excel data can be successfully imported in EG now. ***In order to avoid data truncation during importing, column length needs to be manually modified.***

6. For multi-language Excel files, different language data must be processed respectively according to the steps described above.

Additional font installation

When working with a Unicode environment, additional fonts might need to be installed for Unicode support.

SAS/CONNECT[®] in a Unicode environment

SAS/CONNECT works well in a Unicode environment. When both the SAS server and client are in UTF8 session, the functions provided by SAS/CONNECT, including Data Transfer Service, Remote Compute Service and Remote Library Service, all work well.

Copyright © 2008 SAS Institute Inc., Cary, NC, USA. All rights reserved. SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are registered trademarks or trademarks of their respective companies. All Rights Reserved.