

SAS/ACCESS[®] Interface to PC Files: So Many Options for Microsoft Excel Files — Which Is Best for Me?

Joe Schluter and Henry Feldman, SAS Institute Inc., Cary, NC

ABSTRACT

There are so many ways for SAS/ACCESS[®] users to read and write data from and to Microsoft Excel files: EXCEL, SAS[®] PC Files Server, XLS and XLSX engines, the SAS IMPORT and EXPORT procedures, various Microsoft Excel file formats (.xls, .xlsx, .xlsb, .xlsm), and many more. Many users ask, 'Which is best for me?' This paper explores the requirements and limitations of each engine, along with performance considerations and some not-so-obvious things to consider. It also includes a brief analogous discussion on Microsoft Access databases, which share some of the same mechanisms.

INTRODUCTION

Excel spreadsheets are an immensely popular format to store, process, and share tabular data. Its ad hoc layout and calculation capabilities make it a favorite among analysts. SAS offers an array of SAS/ACCESS engines for reading from and writing to Excel spreadsheets, allowing for seamless integration with other SAS/ACCESS products and related databases.

In the 30+ years since Excel first became available, it has undergone many format and capability changes, affecting the way a client, such as SAS, must interface with Excel. There are also host-specific challenges, such as when accessing Excel from UNIX or Linux. Over the years this led SAS to introduce various engines to the SAS/ACCESS Interface to PC Files product to leverage new capabilities and provide solutions for more customers in various environments.

In this presentation we explore the available SAS/ACCESS engines for Excel, explain some of the inner workings as they relate to the choice of engine to use, and compare them based on these criteria:

- discussion of available SAS/ACCESS engines for Excel files
- host compatibility: Windows versus UNIX or Linux
- Excel file types (.xls, .xlsx, .xlsb, .xlsm)
- full LIBNAME access versus using the IMPORT and EXPORT procedures
- feature richness and functional differences
- international character-set support
- performance considerations
- level of SAS support
- brief discussion of Microsoft Access databases

AVAILABLE SAS/ACCESS ENGINES FOR EXCEL FILES

EXCEL ENGINE

For clarity and ease of reading, EXCEL in all capital letters refers to the EXCEL engine, and Excel refers to an Excel spreadsheet or workbook file.

The EXCEL engine was the first of its kind offered by SAS. It uses the Microsoft Access Database Engine™ (ACE driver) and its OLE DB interface as a client driver. In this way SAS can communicate with the ACE driver using a standard protocol, while the ACE driver handles the details of how to read and write the physical Excel file. While this provides excellent functionality and support for all Excel file types,

its use is limited to SAS running on Windows. Microsoft currently does not provide an ACE driver for UNIX or Linux.

Using the ACE driver also has another noteworthy limitation: Although the ACE driver is available in both 32- and 64-bit libraries (.dll's), the 32-bit version is most often already installed on a Windows PC as part of any Microsoft Office™ installation. Although 64-bit Office is available, most users choose to install the 32-bit version. 64-bit SAS on Windows, however, requires a 64-bit ACE driver. Currently, 32- and 64-bit ACE drivers cannot co-exist on the same system. Therefore, using the EXCEL engine is limited to Windows machines that do not also have 32-bit Office installed.

PC FILES SERVER (PCFS) ENGINE

The PCFS engine bridges some of the limitations of the EXCEL engine. Using a client-server model solves the dependency on SAS running on Windows and also the 32- to 64-bit issue. The server portion of this engine is a stand-alone executable, which is started on a dedicated Windows host. It also uses the ACE driver, but rather its ODBC interface. It is available in both 32- and 64-bit versions and therefore easily communicates with whatever ACE driver bitness might already be present on the server host. The SAS client communicates with the PCFS through a network connection and might therefore be UNIX- or Linux-based.

For SAS users on Windows who are need only work through the 32- or 64-bit ACE driver issue, we provide the PCFS Autostart feature. By omitting the PCFS server name, SAS invokes a PCFS instance with the correct bitness as a background process for the duration of the connection, seamlessly bridging the bitness gap.

As a general rule, PCFS is most often desirable when running SAS on UNIX or Linux or when there is a bitness mismatch between SAS and any existing ACE driver when running SAS on Windows. Users must also consider the slight additional effort required to manage the server itself.

XLS ENGINE

The XLS engine uses a novel approach: Instead of relying on an ACE driver that Microsoft supplies, this engine understands the physical format of .xls files and is therefore host-independent. SAS users on both Windows and UNIX can access this Excel 97-2003 file format as long as the Excel file is accessible through the local or networked file system.

The XLS engine is only available for the IMPORT and EXPORT procedures and is only compatible with the older .xls file format (Excel 97-2003). But it shines with excellent performance because SAS has full control over all components, which allows for the best optimization and also allows SAS to deliver solutions to address some other limitations of the ACE driver.

XLSX ENGINE

The XLSX engine is a close cousin to the XLS engine that specifically supports the newer .xlsx format (Excel 2007 and beyond). It is also a native engine because it understands the physical file format and therefore inherits most of the XLS engine advantages. The XLSX engine also supports full LIBNAME access in addition to the IMPORT and EXPORT procedures.

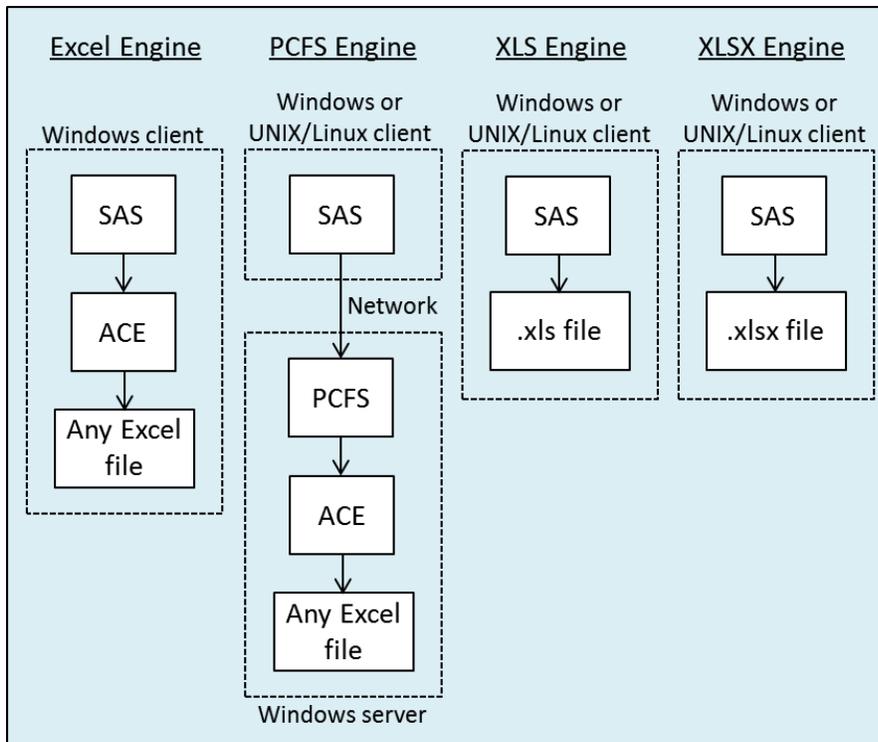


Figure 1. SAS/ACCESS Engines for Excel

HOST COMPATIBILITY

WINDOWS VERSUS UNIX OR LINUX

As previously mentioned, the Microsoft ACE driver is currently available only on Windows. UNIX or Linux users must choose an engine that does not require the ACE driver (XLS and XLSX) or uses a client-server model to bridge that gap (PCFS).

For Windows, the bitness-compatibility of SAS (usually 64-bit) and the ACE driver must be considered. Other software installed, such as Microsoft Office, might control the bitness of the ACE driver, and Windows does not allow for mixed-bitness ACE drivers to co-exist on the same system. In such cases, the PCFS (which might possibly be in Autostart mode), XLS, or XLSX engines should be considered. PCFS is available in both 32- and 64-bit. The XLS and XLSX engines do not require the ACE driver.

If Excel files are not accessible through the user's local or networked file system on the host that is running SAS on Windows, UNIX, or Linux, then PCFS is the logical choice unless another way exists to first copy the Excel file to the client machine, such as using FTP.

EXCEL FILE TYPES

As a result of Excel's long history, these are the Excel file types that are commonly in use.

- .xls Excel 97-2003 workbook
- .xlsx Excel 2007+ XML workbook (the default)
- .xlsb Excel 2007+ binary workbook
- .xlsm Excel 2007+ macro-enabled workbook

SAS/ACCESS engines that use the ACE driver have the advantage of supporting all of these formats both for reading and writing. However, due to ACE driver limitations, creating a new .xlsm format file is not

supported. Creating a new .xlsx format file is possible, starting with the latest release of SAS (9.4 TS1M4).

The XLS and XLSX engines are intended only for the respective formats and do not support .xlsb or .xlsm.

FULL LIBNAME ACCESS VERSUS THE IMPORT AND EXPORT PROCEDURES

Most SAS/ACCESS engines support LIBNAME assignments, providing full DATA step functionality and support for procedures — including WHERE-clause processing, joins, and some use of the SQL procedure. Others support only the IMPORT and EXPORT procedures, which entail a one-time copy operation to transfer an entire table or Excel worksheet to or from a SAS data set or other SAS/ACCESS engine libref.

The EXCEL, PCFS, and XLSX engines support both LIBNAME and the IMPORT and EXPORT procedures. Although the XLS engine supports only the IMPORT and EXPORT procedures, this is not a significant limitation. Due to their nature, most Excel worksheets are fairly reasonable in size. XLS sheets are limited to 65K rows and newer formats top out at around 1 million rows, with the average Excel sheet being in the very low range of these limits. So there is rarely a performance penalty when first importing an entire sheet into a SAS data set and performing any more complex operations on the data in memory. In fact, in many cases this might result in better overall performance because some libref-based DATA step operations might require reading the entire sheet multiple times.

FEATURES AND FUNCTIONAL DIFFERENCES

Many of the available options for LIBNAME and the IMPORT and EXPORT procedures are shared among all four Excel-based engines. Though a full comparison of all options is beyond the scope of this paper, a few have created frequent customer interest and are addressed in more detail.

MIXED DATA TYPES

Compared to other databases, Excel is unusual because it allows each cell to be individually typed, such as number, string, or timestamp. In SAS all values of a given column must be of the same type. Normally, mixed data types are overcome by selecting a column type that allows the representation of all values, typically resulting in conversion to a string.

By default, the ACE driver determines the type of a column by the first non-heading row. This causes a problem if the first data row is numeric, followed by mixed numeric, string, or timestamp data. Cells that cannot be converted to numeric become missing values (“.”), which might be undesirable. Both EXCEL and PCFS engines provide a MIXED= option, but this does not guarantee success, as the ACE driver checks only the first 8 data rows to determine the final type for the column. So if the first 8 rows are numeric, followed by string data, the resulting type is numeric and strings are returned as missing values. There is a Windows Registry entry that you can alter to change this behavior, but it works only for the OLE DB interface (EXCEL engine) but not the ODBC interface (PCFS engine). It is also normally undesirable to modify Windows Registry entries for the benefit of a single application. Even using the DBSASTYPE= option does not alleviate this problem because the ACE driver makes its decision long before this option is applied.

The XLS and XLSX engines do not have this limitation. These engines scan all spreadsheet columns in their entirety to determine the final type of each column. As long as there is a single string-typed cell in an Excel column, the resulting SAS column is of type CHAR. Any additional overhead of this extra step is normally negligible, as Excel spreadsheets are usually not very large.

Of course, any spreadsheet with consistently typed columns is not affected by this.

COLUMN HEADINGS

The ACE driver requires that the very first row of an Excel spreadsheet (or range) provide the column names (which SAS uses) after possible adjustments to avoid unsupported characters, names that are too long, or duplicate names. This makes it impossible to use ACE-based engines (EXCEL and PCFS) to

read spreadsheets or ranges that do not have a heading row. The first row is always used to name the columns and is lost for actual data.

The XLS and XLSX engines provide a GETNAMES=NO option, which creates generic column names (A, B, C, and so on) and reads all requested rows as data.

SUPPORTING OLDER EXCEL VERSIONS AND DRIVERS

Both EXCEL and PCFS engines support a rich set of compatibility options to read and write older versions of Excel files and also a choice of Microsoft client driver.

The VERSION= option determines the type of Excel file to create if the file does not already exist. For reading from an existing Excel file, this is not relevant because the version is automatically detected.

The MENGINE= option selects which Microsoft Access Database Engine™ driver is used. By default, the newer ACE driver is used. However, if it should become necessary to use the older JET driver, this option can be used. Note that the JET driver only supports formats up to Excel 2003 (.xls), and it is not available for 64-bits.

These options are specific to the ACE or JET drivers and therefore are not available for the XLS and XLSX engines. These always create Excel 2003 (.xls) and Excel 2007 (.xlsx) files, respectively.

CONNECTION OPTIONS

The rarely used CONNECT_STRING= option is specific to the ACE or JET driver and is therefore available only for EXCEL and PCFS engines.

INTERNATIONAL CHARACTER SET SUPPORT

Many of our customers around the world require support for non-Latin character sets. In SAS, single-byte character sets (SBCS) allow use of 8-bit ASCII characters, which include English and most Western European languages. However, many Eastern European and Asian languages require extended character sets, such as UTF-8, UTF-16, or language-specific encodings for Chinese, Japanese, or Korean. The multibyte (double-byte character set or DBCS) version of SAS provides support for a wide array of non-ASCII character sets but requires each SAS/ACCESS engine to implement this capability.

Currently, all Excel-based engines provide support for international character sets except for the XLS engine, which is single-byte only. The others base their multibyte support on the version of SAS in use (SBCS versus DBCS). The PCFS, as a stand-alone server, always supports international character sets — using Windows Unicode, as determined by the Windows system locale that is configured on the server — but enables or disables this functionality on a per-connection basis depending on the version of SAS that initiated the connection.

PERFORMANCE CONSIDERATIONS

Most of the time it doesn't matter. Excel spreadsheets tend to be manageable in size. The overhead of any combination of opening and closing the file or starting and connecting to a PCFS outweigh the time spent actually reading the data. However, extremely large spreadsheets — such as in Excel 2007 and above which support over 1 million rows and 16K columns that can result in potentially over 16 billion cells and some operations that require repeated reading of a spreadsheet — might require more planning to optimize performance. Here are some points to consider.

IMPORT DATA INTO SAS FIRST

Take the example of a relatively large Excel spreadsheet on which multiple types of statistical analyses are to be performed in successive DATA steps, procedures, or both. If each step is performed using the libref of a LIBNAME assignment (excluding the XLS engine, which does not support LIBNAME), the entire spreadsheet must be repeatedly read into SAS. You can remedy this by first reading or importing the entire spreadsheet into a SAS data set and then performing the list of functions that you want on the

local copy. Depending on the spreadsheet size, that data can now be held in memory, processed, and analyzed rapidly.

SELECT THE FASTEST ENGINE

Engine selection for Excel files largely depends on other factors, as previously mentioned. If there is a choice and performance is key, the XLS and XLSX engines typically perform much better than the ACE driver-based engines (EXCEL and PCFS). This is because the ACE driver is a generic Microsoft product that supports complex processing such as SQL and handling of numerous connection options. The XLS and XLSX engines bypass the ACE driver and access the Excel file directly, allowing for targeted optimization of these engines for use in the SAS environment.

SQL SUPPORT

SQL is always supported within SAS. However, depending on the database interface (client driver) that you use, SQL can be processed directly in the database by sending it to the client driver, resulting in optimal performance. If the client driver does not support SQL, then SAS reads all relevant data and performs the SQL within SAS.

The ACE driver supports SQL, so complex SQL requests might benefit from using an ACE driver-based engine. XLS and XLSX engines do not directly support SQL, requiring SAS to read all data and perform the SQL processing within SAS, which might reduce performance in some cases. However, most Excel spreadsheets do not contain sufficient data to make this a noticeable difference.

LEVEL OF SAS SUPPORT

SAS Institute is fully committed to support its software in a timely and customer-driven manner. Unfortunately, sometimes issues arise from using products other than SAS — specifically the Microsoft ACE driver in this case. SAS has limited ability to resolve issues that might arise from bugs or limitations in vendor-supplied products, although we try our best to provide coding or usage workarounds through updates and Tech Support. Only the XLS and XLSX engines rely exclusively on in-house SAS code and therefore allow SAS to more quickly and effectively respond to issues that might arise, including the addition of customer-requested features.

MICROSOFT ACCESS DATABASES

In addition to Excel files, the ACE driver supports Microsoft Access databases. Some aspects of this paper might also apply to them. Here is a brief comparison related to the engines discussed so far.

ACCESS ENGINE

The ACCESS engine — not to be confused with SAS/ACCESS, which is a SAS product line — is a close cousin to the EXCEL engine. It also uses the ACE driver locally and must therefore comply with the 32- and 64-bit restrictions we explored earlier. It supports LIBNAME and the IMPORT and EXPORT procedures and is only available for Windows users.

THE PCFS ENGINE

As with the EXCEL engine, the PCFS engine bridges both the bitness and host gaps. It allows reading and writing to Microsoft Access databases from UNIX or Linux, including cases where the existing ACE driver is not bitness-compatible with SAS.

XLS & XLSX ENGINES

These are specific to Excel, and there are no equivalent MDB or ACCDB engines for Microsoft Access databases.

CONCLUSION

This paper has explored various dimensions that determine the requirements and limitations of SAS/ACCESS Interface to PC Files engines that support Excel spreadsheets. In most cases, the choice depends on the type of SAS host (Windows versus UNIX, or Linux), ACE driver bitness, and specific features & performance requirements. With multiple eligible engines, your choice should be based on ease-of-use and speed. Here are some general considerations.

- EXCEL (or ACCESS) engine: Windows only, must be bitness-matched with ACE driver, full SQL support
- PCFS engine: Windows & UNIX/Linux, no bitness requirements, full SQL support
- XLS & XLSX engines: Windows & UNIX, or Linux, no bitness requirements, no direct SQL support, very fast, overcomes some other ACE limitations

Table 1 - Comparison of Engines for Excel Files

Criteria	SAS/ACCESS Interface to PC Files Engine			
	EXCEL	PCFS	XLS	XLSX
Uses ACE driver	Yes	Yes ¹	No	No
ACE driver API used	OLE DB	ODBC	not applicable	not applicable
Host compatibility	Windows	Windows, UNIX	Windows, UNIX	Windows, UNIX
32- and 64-bit requirements	Match ACE	not applicable ²	not applicable	not applicable
Excel file types supported	any	any	.xls	.xlsx
Support LIBNAME	Yes	Yes	No	Yes
Engine name for LIBNAME	EXCEL	PCFILES	not applicable	XLSX
Support the IMPORT and EXPORT procedures	Yes	Yes	Yes	Yes
DBMS= name for the IMPORT and EXPORT procedures	EXCEL	EXCELCS	XLS	XLSX
Support older Excel formats for create	Yes	Yes	No	No
Support CONNECT_STRING=	Yes	Yes	No	No
Support GETNAMES=	No	No	Yes	Yes
Support mixed data in column	Limited ³	Limited ³	Yes ⁴	Yes ⁴
SQL support by client driver	Yes	Yes	No	No
International character set support	Yes	Yes	No	Yes
Performance (large data)	Very good	Good	Best	Best

Table 1. Comparison of engines for Excel files

¹ The ACE driver is used in separate process — possibly on difference machine.

² PCFS bitness must match ACE on the PCFS server machine, not on the SAS client.

³ Mixed data detection: default: 1st row only. When using MIXED= option: 8 rows only.

⁴ Always scans the entire column for the best type.

RECOMMENDED READING

[SAS/ACCESS® 9.4 Interface to PC Files: Reference](#)

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Joe Schluter
SAS Institute Inc.
Joe.Schluter@sas.com
<http://www.sas.com>

Henry Feldman
SAS Institute Inc.
Henry.Feldman@sas.com
<http://www.sas.com>

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 trademarks of their respective companies.