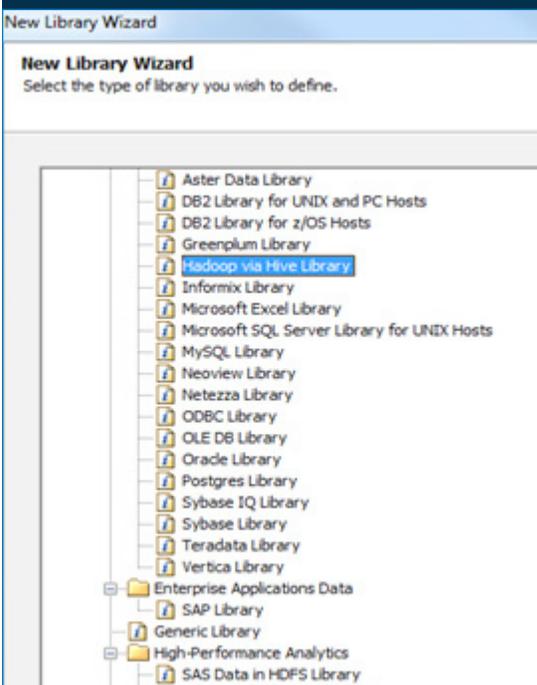


# SAS/ACCESS® Software

Read, write and update data regardless of its native database or platform



## What does SAS/ACCESS® software do?

SAS/ACCESS is an out-of-the-box solution for integrating SAS and third-party databases. You can read, write and update data regardless of the database or platform. Because the data appears native to SAS, there is no need to learn Structured Query Language (SQL) or any other database-specific query languages.

## Why is SAS/ACCESS® software important?

With integration at its source, business decisions are supported by complete, consistent, up-to-date and accurate data. Any source of data can be made available when you need it, plus you can add access modules without replicating or moving the source data.

## For whom is SAS/ACCESS® software designed?

SAS/ACCESS is designed for organizations that need to bring different source system data together into a cohesive and unified environment. It's used by data management professionals and business analysts who are responsible for accessing data in support of applications and business objectives.

With volumes of business information coming from so many sources, it's easy to become overwhelmed with data. Processes slow down. Integrity could be compromised. And, with information stored in disparate environments, you may be less likely to use all your technical resources when running applications - which means you won't get the most value from your data.

SAS/ACCESS software delivers seamless, transparent read and write access to data that's stored on different platforms. It supports the most popular platforms on the market, including Hadoop, open-source platforms, PCs, UNIX and mainframe systems.

With this out-of-the-box technology, data stored in third-party databases can be reached from a common interface as though it were a SAS® data file - so you can analyze or present data directly from the source. Data integrity is not at risk because SAS/ACCESS fully supports the existing security features of the data sources; plus, you can grant access to as many, or as few, users as necessary.

With SAS/ACCESS, you're getting the most up-to-date data possible by reading data directly from the source system into your SAS application. You can also avoid additional source table access overhead by using temporary files, either in the source database or in SAS. And you can use SQL to update, delete or add data to source database files.

## Key Benefits

- **Convenient access to all the data you need.** Seamless and transparent read, write and update rights are provided to more than 60 data sources, including relational and nonrelational databases, PC files, Hadoop, Amazon Redshift and data warehouse appliances.
- **Direct, easy and secure native interfaces.** SQL (or other query language) expertise and custom coding are not required to get a high degree of access to source data. Native security of the target data source is honored, and SAS provides transparent access by making data sources appear as though they are

a native SAS data type, facilitating integration across sources and with SAS procedures, SAS DATA step and SAS solutions.

- **Increased performance and reduced network traffic.** Passing database queries, joins and other functions to the target data source reduces network traffic and speeds data processing times. Faster load times are supported with native bulk-load utilities.
- **Support for integration standards, including ODBC, JDBC (on SAS® Viya®), Spark SQL (on SAS Viya) and OLE DB.** SAS/ACCESS engines are tightly integrated and used by all SAS products and solutions for third-party data integration.
- **Designed to support both technical and business users.** These out-of-the-box access solutions require minimal training and knowledge transfer.

## Product Overview

### Seamless and transparent data integration

With SAS/ACCESS, you can access the most popular databases on common platforms without detailed knowledge of the database or SQL. These engines enable you to treat data as a generally available resource that can be viewed and used regardless of its format and location.

### SQL support

SAS/ACCESS engines support two different methods of native integration: the LIBNAME engine and the pass-through facility. The LIBNAME engine provides a seamless and transparent definition to surface data, defining a simple statement to the source database and requiring minimal knowledge of data or the SQL.

The pass-through facility gives you more flexibility since you can customize SQL statements and modify or keep automatically generated SQL (such as that generated by SAS Data Integration Studio) that's passed directly to the database server. In both methods, SAS statements or functions are mapped to source database-specific statements or functions, and all appropriate SQL statements can be processed directly inside the database for optimal performance.

### Bulk loading and other performance features

SAS/ACCESS supports a range of loading options for moving refined data from SAS into third-party data stores. Bulk loading can significantly enhance performance and be scheduled for off-peak times. Other performance features include a multithreaded read interface to expedite processing, support for automatic partitioning, the ability to pass functionality to the database and the ability to directly control processing.

### Temporary table support

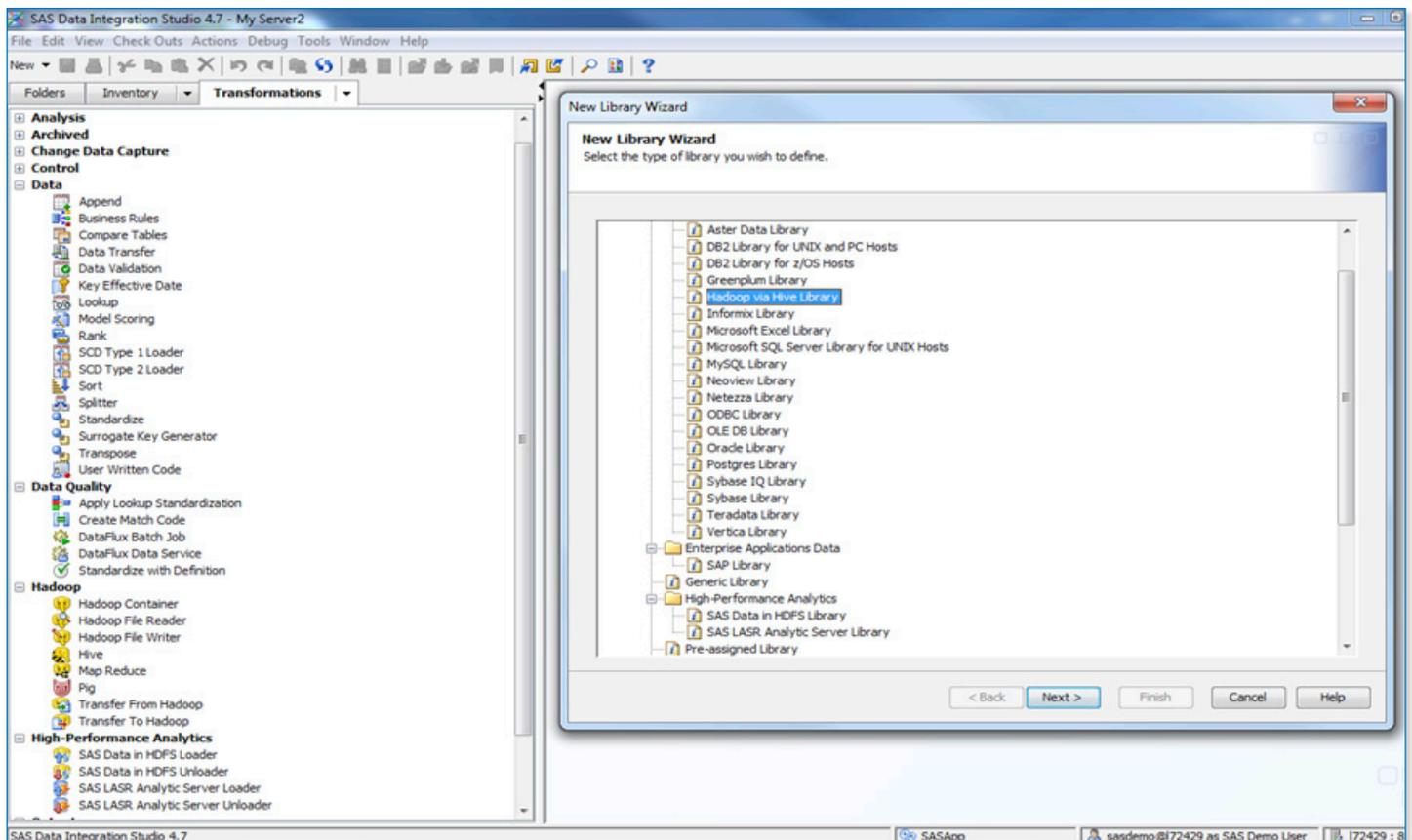
Temporary tables can be created and accessed by multiple SAS processes, providing additional performance gains when the same data is repeatedly required for different steps in a single session. Using temporary tables for heterogeneous joins can also improve performance.

### Metadata integration

DBMS metadata can be accurately maintained within the SAS Metadata Repository. Once you've registered a database table with the SAS Metadata Server, you can use it with other SAS solutions for more efficient processing.

### Data integrity and security

SAS/ACCESS engines accept encoded DBMS passwords so they do not appear as clear text in SAS programs. The software also uses the authentication and login information stored within the SAS Metadata Repository or can use the requirements provided by each database system.



SAS/ACCESS allows you to connect to various data sources and is available via DATA step, LIBNAME statements or through simplified user interfaces such as the SAS Data Integration Studio component of SAS Data Management (shown).

## SAS/ACCESS® for databases

With SAS/ACCESS engines, you can reference database objects directly in a DATA step or in SAS procedures using the SAS LIBNAME statement. You can also use the library window to associate a library reference with relational DBMS objects or a SAS data library. The data values are read as if they were in a SAS data file. You can also use SAS/ACCESS for data warehouse appliances. Typically, these appliances remove complexity of installation with preconfigured database, server and storage in a single unit, all while processing petabytes of data within a highly flexible, scalable environment.

### Supported databases

SAS/ACCESS interfaces to relational databases and database appliances include:

- Teradata
- Aster Data
- Cloudera Impala
- Datacom
- Greenplum
- Netezza
- Vertica
- Informix
- SAP HANA, SAP R3, SAP ASE, SAP IQ
- DB2
- Oracle and Oracle RDB
- PostgreSQL
- Microsoft SQL Server
- MySQL
- Amazon Redshift
- JDBC - SAS Viya
- Spark SQL - SAS Viya
- ODBC
- OLE DB

## Key Features

### Seamless and transparent data access

- Broad access to data through an intuitive interface, regardless of where it's stored. Support for a wide range of databases and platforms, including big data databases, relational stores, data warehouses, mainframe sources and PC files.
- Easy integration with popular platforms without detailed knowledge of the database or SQL.
- Option of working in SAS or SQL with automatic generation of the appropriate SQL statements, passed through to the database for execution.
- Support for integration standards outside of the dedicated SAS/ACCESS module, including SAS/ACCESS Interface to ODBC, SAS/ACCESS Interface to JDBC (SAS Viya) and SAS/ACCESS Interface to OLE DB.
- Ability to execute procedural language, complementing SQL-based statement logic.
- High level of security with native database security mechanisms.
- Option to import or export data from a PC file to a SAS data set, as well as the ability to read and write directly to PC files.
- DBMS metadata can be accurately maintained within the SAS Metadata Repository for metadata reuse.

### Flexible query language support

- Seamlessly access data with minimal knowledge of the data or the SQL required to surface it.
- Take more control by using your own customer SQL statements to modify or maintain automatically generated SQL.
- Map SAS specific statements or functions to database-specific statements or functions and process SQL statements directly inside the database for optimal performance.
- Enlist a SAS extension to process the appropriate query logic using the SAS or database engine.

### Performance tuning options

- Take advantage of a multithreaded read interface, as well as threaded kernel technology and native APIs to Oracle, DB2 Spark and Teradata.
- Enable federated views through optimized read and write, including buffering, compression, threading, chunking, and sort and join performance.
- Join processing is automatically pushed into the database.
- Boost performance with temporary table support.
- Work effortlessly with seamless interfaces to loaders and utilities without an in-depth understanding of each loader.
- Use PROC TRANSPOSE pushdown capability for Teradata and Hadoop.
- Use PROC pushdown capability for Amazon Redshift, Postgres, Microsoft SQL Server and metadata integration.
- Maintain DBMS metadata within the SAS metadata repository, and reuse data jobs.
- Use jobs across a variety of SAS solutions, including SAS® Enterprise Guide® and SAS Data Management.
- Use native storage options, including support for temporary tables, materialized views and partitioned tables.
- Make use of native database types that translate the source database to the appropriate SAS data type.
- Increase database performance with processor threads, placing data into a memory buffer between reads.
- National language support.

## Key Features (continued)

### Improved performance

- Native storage options, including support for temporary tables, materialized views and partitioned tables.
- Native database types translate the source database to the appropriate SAS data type.
- Increase database performance with processor threads, placing data into a memory buffer between reads.
- National language support.

### Optimization features for better performance

- Use pipeline read to increase database read performance by up to 30 percent via a processor thread that reads data from a database and places it into a memory buffer. Pipeline read is available in SAS/ACCESS Interface to DB2 (non-z/OS), SAS/ACCESS Interface to Greenplum, SAS/ACCESS Interface to Oracle and SAS/ACCESS Interface to Teradata.
- Improve efficiency with native storage options, including temporary tables, materialized views and partitioned tables.
- Speed processing with native database types, which are automatically translated to the appropriate SAS data type.

### SAS/ACCESS® for mainframes

SAS/ACCESS for mainframes specifies a view that describes the data source. It also includes descriptors that inform SAS of the information to access and which subsets of data to use. SAS then reads the data values as if they were in a SAS data file. Supported mainframe sources include:

- SAS/ACCESS® Interface to ADABAS
- SAS/ACCESS® Interface to DATACOM/DB
- SAS/ACCESS® Interface to CA IDMS™
- SAS/ACCESS® Interface to IMS-DL/I

### SAS/ACCESS® for NoSQL data platforms

- SAS/ACCESS® Interface to the PI System

### SAS/ACCESS® for distributed file systems

SAS/ACCESS for distributed file systems enables data stored in these source systems to appear as a native SAS data set, easily integrating with other data housed in SAS applications and solutions. The SAS/ACCESS engine also uses the distributed processing capabilities of these systems for even faster processing. One supported distributed file system source is:

- SAS/ACCESS® Interface to Hadoop

### SAS/ACCESS® Interface to PC Files

SAS/ACCESS Interface to PC Files enables native read and write interfacing directly with SAS reports, SAS applications, JMP® and SAS data sets. SAS/ACCESS Interface to PC Files provides direct and highly optimized access to Microsoft Access, Microsoft Excel and data from ODBC data sources on the PC (for example, the Microsoft SQL Server). It also provides access to data stored in SPSS, Lotus Notes, Stata, Paradox and others. Other features include an import/export wizard as well as DBF and DIF procedures.

SAS/ACCESS Interface to PC Files includes access to:

- DBF
- DIF (Unix)
- XLS (Windows)
- WK1
- WK3
- WK4
- XLSX

To learn more about SAS/ACCESS system requirements, download white papers, view screenshots and see other related material, please visit [sas.com/sasaccess](http://sas.com/sasaccess).

To contact your local SAS office, please visit: [sas.com/offices](http://sas.com/offices).

