Configuration Guide for SAS® 9.3 Foundation for UNIX® Environments

Copyright Notice

Table of Contents

Chapter 1 – Introduction

Audience

This document is intended for the SAS Installation Representative, designated as the person responsible for installing and maintaining SAS software for UNIX systems at your site.

This document describes the configuration instructions for SAS 9.3 Foundation, which is made up of server-side Base SAS and a variety of server-side SAS products (the exact products vary by customer). Information about the configuration of mid-tier and client-side products is available from other sources including the SAS Deployment Wizard and any documentation that it might point you to.

The server-side configuration instructions contained in this document are for the configuration of a generic SAS server. If you wish to configure your server for more specific functions, such as a Workspace Server or Stored Process Server, refer to the *SAS 9.3 Intelligence Platform: Application Server Administration Guide* located at `http://support.sas.com/documentation/configuration/index.html`. If you wish to configure your server as an OLAP Server, refer also to *SAS 9.3 Intelligence Platform: Application Server Administration Guide*, at the same location. If you wish to configure your server as a Metadata Server, to the *SAS 9.3 Intelligence Platform: System Administration Guide,* also at the same location.

Understanding This Book

This document conforms to the following conventions:

`Courier`	Courier type indicates commands, directory paths, file names, menu items, Internet addresses, etc.
Italics	Italic type indicates variable text, documentation references, or key notes.
Bold	Bold type indicates important text or concepts.
UPPERCASE	Uppercase type indicates variable and option settings.
Dollar sign $ Pound sign #	A dollar sign $ or pound sign # at the beginning of an example indicates a sample UNIX command line.

Contacting SAS

If you need to contact SAS, refer to the *SAS QuickStart Guide* for contact information.

Accessing Release Documentation

The latest versions of the release documentation are available from the Install Center web page, `http://support.sas.com/installcenter`.

Chapter 2 – Restricted Options

SAS 9.3 Foundation options can be "restricted" by a site administrator so that once they are set by the administrator; they may not be changed by a user. An option can be restricted globally, by group, and by user. To restrict an option it must be added to the appropriate SAS 9.3 Foundation configuration file and this file must have the permissions set by the administrator so that it cannot be updated by users. The option files are processed in the following order: global, group and user. If an option is specified in multiple files, the last occurrence is used.

Global Restrictions

Create the file `!SASROOT/misc/rstropts/rsasv9.cfg` and add options to this file in the normal config file format.

Group Restrictions

Create a file of the following format:

`!SASROOT/misc/rstropts/groups/`*`group-name`*`_rsasv9.cfg`

and add options to this file in the normal config file format.

Example: For user `smith` in the group `staff`: the file name would be `staff_rsasv9.cfg`.

User Restrictions

Create a file of the following format:

`!SASROOT/misc/rstropts/users/`*`user-ID`*`_rsasv9.cfg`

and add options to this file in the normal config file format.

Example:
For user `smith`, the file name is `smith_rsasv9.cfg`.

Additional information

To verify that an option has been set correctly follow this example:

1. Assume the option -EMAILSYS=SMTP was specified in one of the restricted configuration files.

2. Submit the following code:

   ```
   proc options restrict; run;
   ```

   The SAS log should then show a message similar to

   ```
   Option Value Information For SAS Option EMAILSYS
       Option Value: SMTP
       Option Scope: SAS Session
       How option value set:  Site Administrator Restricted
   ```

The following describes the process when a user attempts to change the option value. Assume the option -NOTHREADS was specified in one of the restricted configuration files.

1. Submit the following code:

```
options THREADS;
```

   The SAS log should then show a message similar to

```
   options THREADS;
             -----
              36
WARNING 36-12: SAS option THREADS is restricted by your Site
Administrator and cannot be updated.
```

Note: *Only one Group Restrictions File will be read during SAS processing. The effective groupid of the SAS process that is running is used in the determination of which Group Restrictions File to use.*

Note: *If the effective user ID of the SAS process that is running does not have a corresponding entry in the* `/etc/passwd` *file, then only the global restricted option and the group restricted options files will be read.*

Note: *If the effective groupid of the SAS process that is running does not have a corresponding entry in the* `/etc/group` *file, then only the global restricted option and the user restricted options files will be read.*

Note: *By default, the* `MetadataServer.sh` *script sets the value for the SAS system option MEMSIZE to MAX. Be aware that setting MEMSIZE as a restricted option might cause the metadata server process to fail with memory-related errors. For more information, see Usage Note 43280: Setting MEMSIZE as a restricted option overrides the MEMSIZE setting used by the metadata server, located at* http://support.sas.com/kb/43/280.html.

Chapter 3 – Post-Installation Configuration for SAS Foundation

Configuring Hadoop JAR Files

Updating Your Versioned Jar Repository

If you add SAS/Access Interface to Hadoop to an existing installation, or if new Hadoop JAR files are added or updated, delete the existing `com.sas.app.launcher.cacheFile` file. The file is recreated when SAS is restarted and any new JAR files are discovered.

If default locations were chosen, the file can be found here:

```
$SASHOME/SASVersionedJarRepository/eclipse
```

This step clears the Versioned Jar Repository's cache, so updated versions of Hadoop JAR files (and other updated JAR files) in the VJR are used, rather than the old versions of those JAR files that have been cached.

Note: *This instruction works if Hadoop JAR files have been updated in the Versioned Jar Repository, not in an alternate location.*

Install SAS-required Hadoop JAR Files on Your SAS Machines

SAS components that access Hadoop require that Hadoop JAR files be copied from your Hadoop server onto the SAS machines in your organization that will access Hadoop. Create a directory on your SAS machine that is accessible to all SAS users. For older Hadoop releases (for example, Cloudera CDH3), copy the following Hadoop JAR files into that directory:

- `hive-exec`
- `hive-jdbc`
- `hive-metastore`
- `hive-service`
- `libfb303`
- `pig`
- `hadoop-core`

Newer Hadoop releases split `hadoop-core` into multiple JAR files. For newer Hadoop releases (for example, Cloudera CDH4), copy the following Hadoop JAR files into that directory:

- `hive-exec`
- `hive-jdbc`
- `hive-metastore`
- `hive-service`
- `libfb303`
- `pig`
- `guava`
- `hadoop-auth`
- `hadoop-common`

- `hadoop-hdfs`
- `protobuf-java`

Assistance from your Hadoop administrator may be required to locate the JAR files and network copy them to the SAS machine. Except for `libfb303`, these JAR files include version numbers. For example, on the Hadoop server, the pig JAR file might be `pig-0.8.0`, `pig-0.9.1`, or similar. Do not copy Thrift JAR files such as `libthrift` into the JAR directory.

Set the SAS_HADOOP_JAR_PATH Environment Variable

SAS must be able to find the JAR files. Create an operating environment variable named `SAS_HADOOP_JAR_PATH` as the directory path of the JAR files. For example, if the JAR files are copied to directory `/users/third_party/Hadoop/jars`, then the following command sets the environment variable appropriately:

```
export SAS_HADOOP_JAR_PATH=/users/third_party/Hadoop/jars
```

Set `SAS_HADOOP_JAR_PATH` in a permanent manner for all SAS users who access Hadoop from this machine.

A `SAS_HADOOP_JAR_PATH` directory must not have multiple versions of a Hadoop `.jar`; this can cause unpredictable behavior in SAS.

***Note**: For the SAS/ACESS Interface to Hadoop to operate properly, your `SAS_HADOOP_JAR_PATH` directory must not contain any Thrift jars such as `libthrift*.jar`.*

JAR Files for MapR-based Hadoop Systems

Along with the documented JAR files (`hive-*.jar`, etc.), you also need to point to the JAR files provided in the MapR client installation.

For example,

```
export
SAS_HADOOP_JAR_PATH=/users/third_party/Hadoop/jars;/opt/mapr/hadoop/ha
doop-0.20.2/lib
```

where `/users/third_party/Hadoop/jars` is as described above, containing `hive-*.jars`, etc., and where `/opt/mapr/hadoop/hadoop-0.20.2/lib` is the JAR directory laid down by the MapR client installation software.

In addition, SAS must be pointed to the MapR client installation directory that contains the MapRClient sharable library (for example, on Linux , the `libMapRClient.so`):

```
SAS-Invocation -jreoptions (-Djava.library.path /opt/mapr/lib)
```

In most installations, the `-jreoptions` addition would be placed in the site-specific SAS configuration file.

Supporting Multiple Hadoop Versions and Upgrading Hadoop Version

The JAR files in the `SAS_HADOOP_JAR_PATH` directory must match the Hadoop server to which SAS connects. If you have multiple Hadoop servers running different Hadoop versions, then create and populate a separate directory with version-specific Hadoop JAR files on the SAS machine for each Hadoop version. `SAS_HADOOP_JAR_PATH` then must be dynamically set depending on which Hadoop server a SAS job or SAS session will connect to. One means to

dynamically set `SAS_HADOOP_JAR_PATH` is to create a wrapper script associated with each Hadoop version. SAS is invoked via a wrapper script that sets `SAS_HADOOP_JAR_PATH` appropriately to pick up the JAR files that match the target Hadoop server.

Upgrading your Hadoop server version may involve having multiple Hadoop versions active. The same multi-version instructions apply.

Chapter 4 – Post-Installation Configuration for User Authentication and Identification

Overview

UNIX user security is more than just authentication. User identification is also performed when user credentials are validated. Unlike Windows, UNIX uses an integer value, called the UID, to identify users. Ownership of system resources is then assigned by associating a particular UID with a system resource. User identification determines the UID for a particular user name.

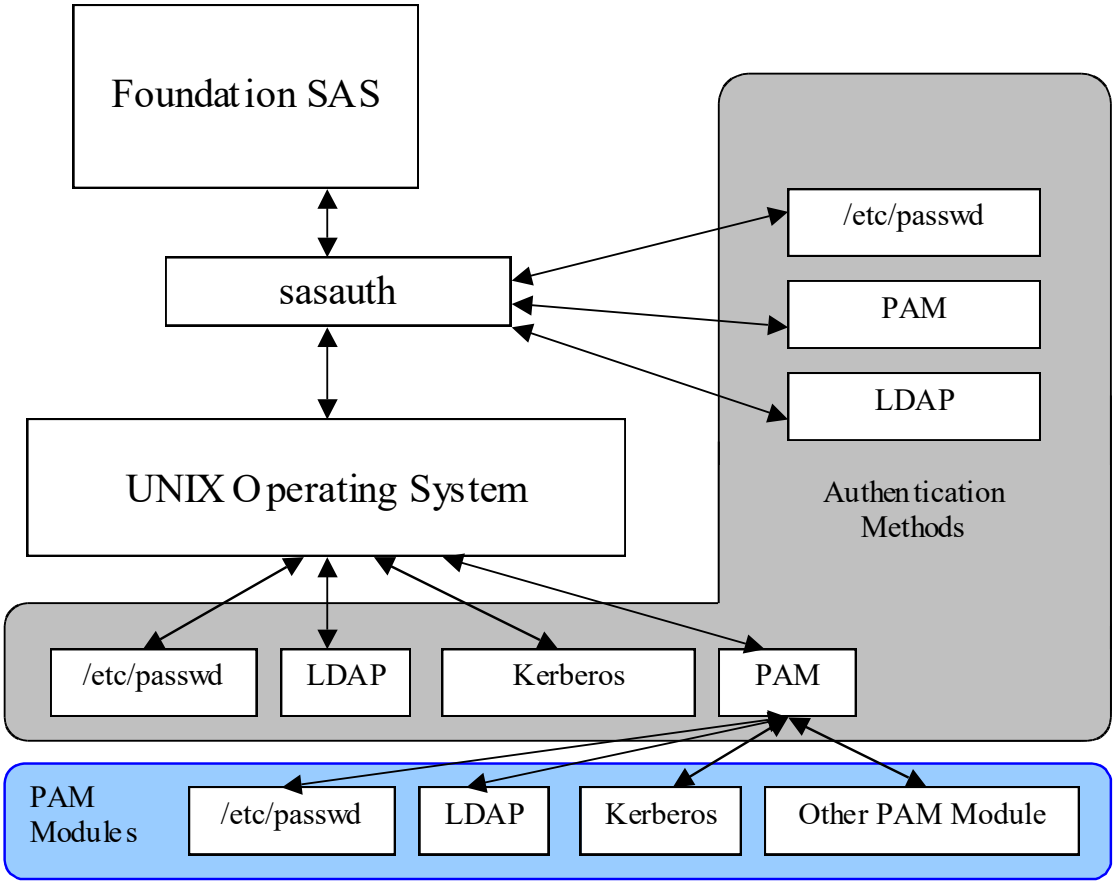
When user credentials are validated, UNIX systems will search a user database for an entry that contains the same user name. Traditionally, the user database is plain file in the file system, but newer security environments may store this in a binary database, or on a server on the network. Most UNIX systems support several other storage methods than the traditional file. Once the user entry is found, the password can be retrieved and matched against an encrypted version of the user provided password (authentication), and the UID is retrieved for the user (identification).

SAS for UNIX systems validates users in the same manner. The user name provides an index into a user database, from which the user is identified and authenticated. Usually superuser permissions are required to read the user database. Since running the entire SAS process with superuser permissions is undesirable – users would have access to files they don't own – an external utility, named sasauth (found in `!SASROOT/utilities/bin`), is used to perform authentication. The sasauth process runs `setuid` to root, so that it has the appropriate permissions to access the user database.

Authentication databases can be stored in several places. The traditional form is as a text file, `/etc/passwd`, with encrypted passwords stored in `/etc/shadow`. Newer forms utilize client/server architecture to provide network-wide authentication, as in NIS+ and LDAP.

For each of these forms, the operating system or application that performs user credential validation must implement the necessary functionality to access the database. Since each form has a different application interface, it is very difficult to support all authentication forms. PAM, or pluggable authentication modules, is a standard library for performing user authentication (but not identification). PAM uses "modules", or libraries, to access multiple authentication forms. A system administrator can select the appropriate authentication based on security requirements. Most UNIX systems support PAM in addition to the native operating system authentication.

The following figure shows possible authentication flows.



SAS strongly recommends that the base operating system be configured to use the required authentication/identification form that matches local requirements. For example, if the SAS server is installed at a site where there is a central LDAP repository, the operating system should be configured as an LDAP client for the central repository.

Many sites like to use PAM since it is a widely accepted authentication mechanism and is very flexible. Modules can be obtained for custom authentication mechanisms, such as smart cards, and added to the system without direct application support. But PAM's lack of user identification is problematic for use with sasauth. The PAM programming libraries will only authenticate a user/password combination. The UID, which is needed by SAS, is not returned. Therefore sasauth will use the standard UNIX authentication calls to obtain the UID, meaning that the system must also be configured to access the same user information as PAM. If you find that your site needs to use PAM for authentication, configuration instructions are provided in subsequent sections of this document.

Configuring User Authentication

Certain SAS products and features employ functionality that require SAS to check user ID authentication and file access authorizations. This in turn necessitates that certain files within your SAS installation have setuid permissions and be owned by root. Configuring user authentication is required for all users of SAS software. You can perform this by issuing the following commands at the UNIX command prompt.

```
$ su root
# cd !SASROOT/utilities/bin
# mv setuid/* .
# chown root elssrv sasauth sasperm
# chmod 4755 elssrv sasauth sasperm
# exit
```

Configuring sasauth

sasauth natively supports system authentication (such as `/etc/passwd`), LDAP repositories, and PAM authentication. It also provides three levels of logging, and user retry lockout, where a user will not be allowed to authenticate for a certain period of time after a certain number of invalid authentication attempts are made. All of these features are configured via a text file, `!SASROOT/utilities/bin/sasauth.conf`.

The sasauth configuration file is a text file consisting of name/value pairs for configuring behavior, one per line. Names and values are case sensitive. A "#" character is used for comments, which extend to the end of the line.

Supported names and values are listed below.

Name: methods

The methods setting specifies what user validation methods should be used. At least one should be specified, though multiple values may be specified, separated by spaces. Authentication then follows the listed methods in order from left to right, which each method attempted until the user identity is found.

Values	Usage
`pw`	Use system authentication, typically `/etc/passwd` - `/etc/shadow` authentication. On some hosts, this also includes protected password databases or OS-provided enhanced security.
`pam`	Use PAM for authentication. The operating system's user security functions are also used to determine the user's UID and GID. PAM must be configured properly for sasauth, as described in "Configuring PAM Authentication for Use with sasauth" below.
`ldap`	Use LDAP queries for authentication. See "Using the sasuath LDAP Authentication Method" below.
`ext`	Use a custom authentication mechanism. This mechanism is built using the sasauth customization kit, which can be found in `!SASROOT/utilities/src/auth`.

Name: debugLog

Name: accessLog

Name: errorLog

These settings specify the pathnames for the sasauth logs. sasauth provides three logs:

- error log – Contains error messages.
- access log – Contains transaction information for each user validation request: user name, validation method used, and validation result.
- debug log – Contains verbose debugging information. Useful when troubleshooting initial configuration.

The value should contain the path for the log file. Log files whose paths are unspecified are not generated, with the exception of the error log, which is sent to syslog instead.

Log file paths may not include system directories (e.g. /dev, /usr, /etc). If a log path contains a system directory, sasauth will not create the log and will write a message to the error log or syslog.

For example:

```
#debugLog=
accessLog=/tmp/sasauth.log
#errorLog=
```

will configure sasauth to have no debug log, use `/tmp/sasauth.log` as the access log, and syslog for the error log.

Note: *You may need to configure your system's syslog facility to see sasauth messages. Refer to your system documentation for details.*

Name: logOwner

Specifies the numeric UID of the owner of the sasauth log files. Defaults to root since sasauth runs as root. Use this setting to allow a user other than root to read the sasauth log files.

Name: debugNoPasswords

When set to "true", passwords will not be written to the log file. Defaults to true.

Name: maxtries

Name: maxtriesPeriod

Name: maxtriesWait

These settings configure sasauth's maxtries configuration. sasauth will not authenticate a user after a maximum number of attempts are made in a given period of time. The user must then wait for a given wait time before additional authentication requests are validated. When maxtries is activated, information about maxtries failures is logged to the access log. The setting maxtries is the maximum number of attempts that may be made. maxtriesPeriod specifies the number of seconds after which repeated attempts exceeding the maxtries count are not authenticated. maxtriesWait specifies the number seconds the user must wait before the maxtries count is reset and validation requests are then permitted.

For example, these settings:

```
maxtries=5
maxtriesPeriod=60
maxtriesWait=300
```

will cause sasauth to stop authenticating a user for 5 minutes if 5 invalid attempts are made in 1 minute.

To turn off maxtries, remove all three settings from the configuration file by commenting them out.

AIX-specific Options for Password Validation

The following options for AIX and instruct sasauth to use AIX-specific system calls when validating credentials using the "pw" authentication method.

Name: AIX_LOGIN_CHECK

If TRUE, check S_LOGINCHK flag when authenticating. If unspecified, default value is TRUE. The option must be explicitly set to FALSE to bypass the check.

Setting this value to true, will allow system administrators to block access to SAS servers and services for users by changing the value for S_LOGINCHK.

Setting this value to false, which bypasses the check, allows system administrators to turn off interactive logins but still allow users to utilize SAS servers and services.

Name: AIX_REPORT_RESULT

If TRUE, sasauth will report the result of authentications via the "pw" authentication method to the operating system. Defaults to FALSE.

Setting this value to true allows SAS authentications to be tracked in the /etc/security/lastlogin database. Information for SAS authentications is shown with the TTY name "SAS".

Name: AIX_USE_AUTHENTICATE

If TRUE, sasauth "pw" authentication method will use the AIX system subroutine authenticate() to validate user credentials instead of the traditional UNIX algorithm for user validation.

Note: *The AIX authenticate() routine does not distinguish between an expired password and a bad password. When using the authenticate routine, sasauth will never return "account expired" for expired accounts/passwords. Users will always get a more generic "authentication failed" message.*

Using the sasauth LDAP Authentication Method

The sasauth LDAP authentication method ("method=ldap" in the configuration file) provides a direct connection from sasauth to an LDAP database for authentication. Connections from sasauth to LDAP servers will be encrypted if specified in the sasauth configuration file. sasauth will query user attributes from the database and then authenticate the user based on the returned attributes. sasauth will also query the LDAP database to determine secondary group attributes for the user being authenticated.

LDAP repositories that are used for UNIX authentication (including sasauth) must include UNIX/Posix user attributes (such as UID) in the database. Without this information, the LDAP database cannot be used with UNIX. Most LDAP servers provide an LDAP schema that contains this information. Microsoft Active Directory repositories should have Microsoft Services for UNIX (SFU) 2 or 3 installed. Other LDAP databases should conform to the RFC 2307 standard for including UNIX user attributes in an LDAP database. sasauth requires the following user attributes, listed using their RFC 2307 names:

- uid - user name

- uidnumber - numeric UID
- gidnumber - numeric group number of the user's primary group
- userpassword - encrypted form of user's password. sasauth supports crypt, SHA, and SSHA forms.
- shadowLastChange - date of last password change
- shadowMax - Maximum age of password before change is required
- shadowExpire - Expiration date of account.

Note that password expiration will not be processed by sasauth if the password expiration attributes are not found in the database.

sasauth also requires the following group attributes, listed using their RFC 2307 names:

- group - group name
- gidNumber - numeric ID of the group
- memberUid - user name that is in the group

The memberUid attribute is repeated for each member of the group.

Configuring the sasauth LDAP Authentication Method

Once the LDAP method is added to the list of authentication methods for sasauth (see "Name: methods" above), additional settings will need to be configured for LDAP in sasauth.conf. The names and values are listed below.

Name: LDAP_HOST

Name: LDAP_PORT

Name: LDAP_SSL_HOST_PORT

Host name, port number, and LDAP/s port number of the LDAP server. LDAP_PORT and LDAP_SSL_HOST_PORT can be omitted, in which case sasauth will use the standard LDAP port number. sasauth will use LDAP_SSL_HOST_PORT instead of LDAP_PORT if encrypted communications are active. (See the setting “LDAP_BIND_SECURITY” below.)

NAME: LDAP_HOST_LIST

Specifies a list of LDAP hosts to use. Entries in the list are separated by spaces, and are of the form “hostname:portnumber”. Port number can be omitted to use the standard port number or standard LDAP/s port number. For example:

```
LDAP_HOST_LIST=host1 host2.mycompany.com:3000
```

Hosts are queried from left to right. Hosts are not used if a network connection cannot be made. If a connection is successful, then that host is used for LDAP queries.

Name: LDAP_AUTH_METHOD

Name: LDAP_HOST_DN

Name: LDAP_HOST_PW

Name: LDAP_GROUP_METHOD

sasauth will authenticate user credentials by using bind or match. For bind, sasauth will bind to the server with the user's credentials. If the bind fails, the user is not authenticated. By binding to

the server using the user's credentials, the LDAP server does all of the authentication (including applying security rules not supported by sasauth), but sasauth cannot determine the specific cause of failed logins. Users will not know why authentication failed when using bind for authentication.

To use bind authentication, set LDAP_AUTH_METHOD to the value BIND (case-sensitive) in the configuration file.

For match, the user's encrypted password and expiration information are queried from the database and matched with the provided credentials. A mismatch or expiration causes the authentication to fail.

To use match authentication, set LDAP_AUTH_METHOD to the value MATCH, and set LDAP_HOST_DN and LDAP_HOST_PW to the user and password for an admin user. An admin user is required because LDAP will not return the encrypted password to a non-administrative user. Since the `sasauth.conf` file will now contain password information, make sure that it is readable only by root (for example, run `chmod 400 sasauth.conf` from the shell).

LDAP_GROUP_METHOD controls how sasauth will bind when querying secondary group memberships (the LDAP equivalent of reading /etc/group) from the LDAP server. When set to USER, sasauth will bind using the user's credentials. When set to HOST, sasauth will use the credentials specified for LDAP_HOST_DN and LDAP_HOST_PW when binding to the LDAP server. Use the "HOST" setting if users do not have sufficient access privileges to read group membership information.

Name: LDAP_BIND_SECURITY

Specifies the security/encryption used when binding to the server. Use the value "simple" for standard LDAP authentication. Use the value SSL for encrypted communications via LDAP/s. Defaults to "simple".

When set to SSL, the system must install security certificates and configure sasauth to use them. See "Installing and Configuring LDAP/s Certificates" below.

Note: *Encrypted communication via LDAP/s is not supported on Linux.*

Name: LDAP_SEARCHBASE

Name: LDAP_USERBASE

These settings provide the search criteria used by sasauth when constructing queries to retrieve user identification. For example:

```
LDAP_SEARCHBASE="DC=MYGROUP,DC=MYCOMPANY,DC=COM"
LDAP_USERBASE="ou=People"
```

Set to values appropriate for your organization. Your LDAP administrator can assist with determining these values.

Name: LDAP_USERFILTER

Specifies a filter clause used while authenticating that limits access to SAS servers and services.

For example:

```
LDAP_USERFILTER="(gidNumber=100)"
```

would result in an LDAP query that returns no results for users not in group 100, limiting access to only users in group 100.

Name: LDAP_IGNORE_USERNAME

When set to TRUE, sasauth will ignore domain specifications in usernames and pass them to the LDAP server unmodified. When unset, the domain is extracted and an extra OU clause is added containing the domain.

For example:

```
fred@purchasing or purchasing\fred
```

results in:

`msSFU30Name=fred,ou=purchasing,dc=company,dc=com` (option unset)

`msSFU30Name=fred@purchasing,dc=company,dc=com` (option set)

This setting is helpful when working with Active Directory as your LDAP database.

Name: LDAP_SCHEMA

Specifies which schema the server uses. Select from:

- `LDAP_SCHEMA=RFC2307` - for RFC 2307 (e.g. Sun ONE Directory Server),
- `LDAP_SCHEMA=AD2` - for Active Directory with Services for UNIX (SFU) 2
- `LDAP_SCHEMA=AD3` - for Active Directory with Services for UNIX (SFU) 3
- `LDAP_SCHEMA=OTHER` - for a manual configuration. Follow instructions in the configuration file when using this value.

Installing and Configuring LDAP/s Certificates

When using LDAP/s, security certificates must be installed on the system. sasauth uses the standard system SSL libraries, so certificates are installed using operating system utilities. General instructions for installing certificates for each UNIX environment are given below. LDAP servers may require more than one certificate, usually a "root" certificate for your site and a server certificate for the LDAP server itself.

Note: *The following examples use a certificate in binary (.cer) format.*

When installing certificates, order is usually specific. First install your "root" certificate and then additional certificates for the servers LDAP will be accessing.

Solaris and HP-UX Certificates

These hosts use the certutil utility to import certificates. It has the path:

`/usr/sfw/bin/certutil` (Solaris)

`/opt/ldapux/contrib/bin/certutil` (HP-UX)

certutil will read the certificates and add them to the certificate database. The certificate database is usually located in:

`/var/ldap` (Solaris)

`/etc/opt/ldapux` (HP-UX)

Your system administrator can place the certificate database in an alternate location, but leaving them in the standard location will make them available to other applications on the system that use the system's version of the LDAP libraries.

Install the certificates as follows. Root permissions are required.

1. Create the certificate directory if it doesn't exist.
   `mkdir /var/ldap` (Solaris)
   `mkdir /etc/opt/ldapux` (HP-UX)
2. Import the certificates. The `-n certutil` option specifies the name of the certificate, and should match the name encoded inside the certificate.

   ```
   certutil -A -a -i rootcertificate.cer -n "Root CA" -t "CT" -d
   /var/ldap (Solaris)
   certutil -A -a -i server.cer -n "ldapserver" -t "CT" -d
   /var/ldap (Solaris)


   certutil -A -a -i rootcertificate.cer -n "Root CA" -t "CT" -d
   /etc/opt/ldap (HP-UX)
   certutil -A -a -i server.cer -n "ldapserver" -t "CT" -d
   /etc/opt/ldap (HP-UX)
   ```

3. Modify the permissions of the certificates so all users can read them.

   ```
   chmod 644 /var/ldap          (Solaris)
   chmod 644 /var/ldap/*.db

   chmod 644 /etc/opt/ldapux    (HP-UX)
   chmod 644 /etc/opt/ldapux/*.db
   ```

4. Validate the certificates using the list option (-l) of certutil.

   ```
   certutil -L -d /var/ldap     (Solaris)
   certutil -L -d /etc/opt/ldapux (HP-UX)
   ```

Once the certificates are installed, sasauth.conf can be changed to match the installed certificates. Certificate settings are at the end of the sasauth.conf file. Solaris and HP-UX require the following settings.

Name: LDAP_SSL_CERTIFICATE_FILE

Specifies the path/file name for the certificate database. On HP-UX and Solaris, this should specify the directory that contains the certificate files, otherwise sasauth will get "Bad database" errors when initializing SSL.

Name: LDAP_SSL_STRENGTH

Specifies how the certificates will be validated. Select from:

- `LDAP_SSL_STRENGTH=CERT` – Accepts the server's certificate only if certificate authority is trusted.
- `LDAP_SSL_STRENGTH=WEAK` – Accepts the server's certificate without validating certificate authority.

- `LDAP_SSL_STRENGTH=CNCHECK` – Same as CERT, but match the CN attribute with the server's DNS name. With this value set, LDAP_HOST_LIST may not be used.

The value "CERT" is used in most cases.

AIX Certificates

AIX certificate management tools are provided in the IBM Global Security Kit (GSKit). Use of the kit is documented in section 4.3.1, "Configuring SSL", of the IBM Redbook "Integrating AIX into Heterogeneous LDAP Environments." The Redbook is available at http://www.redbooks.ibm.com/redbooks/pdfs/sg247165.pdf.

GSKit provides the command "gsk7cmd" to create and maintain SSL certificates. (There is also a graphical tool, gsk7ikm, that can be used. The examples that follow use the command line tool.) If the utility is not available on your system, then you need to install the requisite packages as described in the Redbook.

Certificate files (also called key files or the key database) are commonly created in `/etc/security/ldap`, but that directory also contains many other files used by the AIX LDAP client software. Your administrator may want to add use `/etc/security/ldap/keys` instead.

Install the certificates as follows. Root permissions are required.

1. Create the directory if it doesn't exist.

```
mkdir /etc/security/ldap
```

2. Create a key database. The –pw option is the password for the certificate database. Select a password appropriate for your site.

```
gsk7cmd -keydb -create -db /etc/security/ldap/key.kdb -pw
ls93key -type cms
```

3. Import your certificates. The –label option is a symbolic name used to identify the certificate in the database. Select a name that uniquely identifies the certificate.

```
gsk7cmd -cert -add -db /etc/security/ldap/key.kdb -pw ls93key
-file rootcertificate.cer -format ascii -label "Root CA"
-trust enable
gsk7cmd -cert -add -db /etc/security/ldap/key.kdb -pw ls93key
-file server.cer -format ascii -label "ldap server" -trust
enable
```

4. Test your database by listing the contents. You should see all of the system certificates and the certificates you just added.

```
gsk7cmd -cert -list CA -db /etc/security/ldap/key.kdb -pw
ls93key
```

5. Validate the permissions for the new certificates. Check the directory that contains the certificates and the files themselves. All users should have read permission for the files and read/execute permissions for the directory.

```
ls -l /etc/security
ls -l /etc/security/ldap
```

Once the certificates are installed, sasauth.conf can be changed to match the installed certificates. Certificate settings are at the end of the sasauth.conf file. AIX requires the following settings.

Name: LDAP_SSL_CERTIFICATE_FILE

Specifies the path/filename for the certificate database. On AIX, this is the full path for the key.kdb, as was used with the gsk7cmd utility.

Name: LDAP_SSL_CERTIFICATE_NAME

The name/alias of the certificate to be used when connecting with the LDAP server, usually your root certificate. This should be the name specified in the certificate. You can determine the name of the certificate by executing the command:

```
gsk7cmd –cert –details –db /etc/security/ldap/key.kdb –pw ls93key
–label “Root CA”
```

The name is found in the “Subject:” field of the command output.

Name: LDAP_SSL_CERTIFICATE_PASSWORD

The password for the certificate file, as specified when using the gsk7cmd above. Quotes are not necessary.

Note: *Since the sasauth configuration file now contains a password, check the permissions on the sasauth.conf file. It should only be readable by root.*

Example: sasauth.conf Settings for LDAP/s

The following are the necessary settings for sasauth to use encrypted communications with a Sun Directory Server on AIX without including an LDAP bind password. Only members of group 112 will be able to connect.

```
methods=ldap
LDAP_HOST=ldap.company.com
LDAP_AUTH_METHOD=BIND
LDAP_GROUP_METHOD=USER
LDAP_BIND_SECURITY=SSL
LDAP_SEARCHBASE=”DC=group,DC=company,DC=com”
LDAP_USERBASE=”ou=People”
LDAP_USERFILTER=”(gidNumber=112)”
LDAP_SCHEMA=RFC2307
LDAP_SSL_CERTIFICATE_FILE=/etc/security/ldap/key.kdb
LDAP_SSL_CERTIFICATE_NAME=”Root CA”
LDAP_SSL_CERTIFICATE_PASSWORD=ls93key
```

Configuring PAM Authentication for Use with sasauth

PAM is architected such that applications must be registered in order to use authentication services. For sasauth to perform authentication, entries must be made in the PAM configuration that describe what authentication services are used when sasauth performs an authentication, specifically the “account” and “auth” module types.

Note: *PAM allows configuration of “other,” which permits any application to use authentication services. This is not recommended.*

PAM supports applications that run in both 32-bit and 64-bit environments. Modules used with sasauth must match the binary format of the sasauth program. For SAS 9.3 on UNIX platforms, sasauth is a 64-bit binary, and PAM modules must be 64-bit libraries. The standard system modules are usually provided in both 32-bit and 64-bit versions, with each set stored in a

separate directory. pam.conf then contains pathnames that are either relative (Solaris and AIX) or contain a symbolic variable (HP-UX) that allows the correct format to be loaded depending on the format of sasauth.

On HP-UX, Solaris, and AIX systems, the PAM configuration is stored in `/etc/pam.conf`. For sasauth authentication to succeed, entries should be added of the following form:

```
service-name module-type control-flag module-path options
```

For example, these entries enable sasauth to authenticate on Solaris:

```
sasauth auth requisite      pam_authtok_get.so.1
sasauth auth required       pam_dhkeys.so.1
sasauth auth required       pam_unix_auth.so.1
sasauth account required    pam_unix_account.so.1
```

To authenticate on HP-UX:

```
Sasauth account required    /usr/lib/security/$ISA/libpam_unix.so.1
Sasauth auth required       /usr/lib/security/$ISA/libpam_unix.so.1
```

If the system uses an authentication service other than the UNIX password files (such as LDAP or Kerberos), then the entries will have to define what service to use. The manual page for `/etc/pam.d` will help determine these entries.

On Solaris, if LDAP is being used, PAM should also be configured to communicate with the directory server via the `ldapclient(1m)` command. Refer to the ldapclient man page for more information.

Note: *AIX systems do not ship with PAM activated. Refer to the IBM* document Security Guide – Authentication Module *(http://www16.boulder.ibm.com/pseries/en_US/aixbman/security/pam_overview.htm) for instructions on activating PAM on AIX.*

On Linux systems, the directory `/etc/pam.d` contains a file for each program authorized to use PAM. The name of the configuration matches the name of the process making authentication requests. For sasauth, the configuration file is `/etc/pam.d/sasauth`.

The configuration file contains entries in the following form:

```
module-type control-flag module-path options
```

For example, `/etc/pam.d/sasauth` may contain:

```
#%PAM-1.0
auth    required        pam_unix2.so    nullok
account required        pam_unix_acct.so
```

Note: *In the SAS Intelligence Platform, PAM is an optional configuration that is useful only in certain circumstances. For guidance and alternatives, see the discussion of authentication in the* SAS Intelligence Platform: Security Administration Guide.

AIX: Using System LDAP Authentication with sasauth

IBM does not provide an LDAP module for PAM. The open source package OpenLDAP can be used to build an LDAP module, but this is not recommended for production environments since it is not a solution supported by IBM. Instead, sites that need LDAP authentication should configure the AIX system for LDAP authentication. Refer to the IBM Redbook *Integrating AIX into Heterogeneous LDAP Environments* for instructions on how to configure AIX as an LDAP client.

Solaris: LDAP and Numeric User Names

The Solaris LDAP client does not treat numeric user names as user names. Instead, Solaris assumes that a user name that is numeric is actually a UID, and converts the user name directly to the UID instead of querying the LDAP database. Since Solaris recommends that user names begin with an alphabetic character, this is unlikely to change. If your site uses Solaris as an LDAP client, then user names in LDAP cannot be numeric.

Customizing Authentication and Identification

sasauth can be configured to perform authentication in a site-specific manner. The SAS Foundation installation includes the *UNIX Authenication API*, a package for developing site-specific authentication and identification.

The files and documentation are installed in `!SASROOT/utilities/src/auth`. Refer to the file `docs.pdf` in that directory for detailed development instructions.

Chapter 5 – Configuring Integrated Windows Authentication

Integrated Windows Authentication (IWA) configures participating SAS servers to accept users who have successfully authenticated to their Windows desktop. It is primarily used for connections to the metadata server and the standard workspace server, but it is also supported for direct connections to an OLAP server (for example, from a data provider). Username/password authentication takes place using PAM in the default IWA setup, so PAM configuration is required (for more information, see the preceding chapter, "Post-Installation Configuration for User Authentication and Identification").

There are a number of benefits associated with IWA:

- Bypasses the initial logon prompt.
- Accommodates logon mechanisms that are not password-based (such as smart cards or biometrics).
- No user credentials are transmitted.
- Uses the Kerberos protocol which relies on exchanging tickets rather than passwords – this process happens automatically without user knowledge.
- Clients can talk to Windows and UNIX servers (see the limitations listed below).
- Users don't need the **Log on as a batch job** privilege.

There are also some significant limitations that should be accounted for:

- All participating clients and servers must authenticate against the same Windows domain (or against domains that trust one another).
- Web applications can't participate in this implementation of IWA. However, if you configure Web authentication, and your Web environment offers IWA, then your Web applications can use IWA. SAS provides instructions for configuring IWA for your Web application server at `http://support.sas.com/thirdpartysupport`.
- If you use IWA for the metadata server, there are no cached credentials from an initial logon. For this reason, it is a good idea to configure IWA for the workspace server also.
- Desktop clients that run on UNIX (for example, SAS Management Console on UNIX) can't participate in IWA.
- In order to use IWA on UNIX, you must purchase, install, and configure an additional third-party product (Quest Authentication Services 4.0).
- When you use IWA on UNIX, only Kerberos connections are supported (there is no support for NTLM on UNIX).

The use of Integrated Windows Authentication is optional.

Prerequisites for Integrated Windows Authentication on UNIX

To use IWA for a server on a UNIX host, you must complete the following prerequisite steps:

1. Purchase, install, and configure Quest Authentication Services. Verify that your UNIX host has joined the Active Directory domain and is represented in Active Directory as a computer object.

Note: *In the initial release of SAS 9.3, the only supported implementation of IWA on UNIX requires Quest Authentication Services 4.0.1.23 (or later).*

2. Create a service account and corresponding keytab file. For example, on your UNIX host, from `/opt/quest/bin/vastool`, run the following command:

```
vastool -u admin service create SAS/
```

Here are some details:

- vastool is a command line utility that lets you manage your Quest Authentication Services deployment, information in Active Directory, keys, and Kerberos tickets.
- In the `-u` option, specify an Active Directory identity under which vastool can connect to Active Directory and create users. You will be prompted for the password.
- In this example, `SAS` is the service class name of the service account that is created. You must use this service class name in order to create the default service principle name (SPN) that clients expect.
  Note: *The alternative, using a custom SPN, can be labor-intensive and error-prone.*
- The command creates a service account in the default Computers container on the Active Directory domain. The account name is in a format such as *machine-service*. In this example, if the UNIX host is **machineA.unx.company.com**, then the service account name is **machineA-SAS**. A random password is generated for the account.
- The service has an associated user principal name (UPN) in the format *account-name@Kerberos-realm*. In this example, the Kerberos realm is **COMPANY.COM**, so the UPN is `machineA-SAS@COMPANY.COM`.
- The service also has an associated service principal name (SPN) in the format *service/machine@Kerberos-realm*. The machine value must be specified as a fully qualified domain name (FQDN). In this example, the SPN is **SAS/machineA.unx.company.com@COMPANY.COM**.
- The service has a corresponding Kerberos keytab file. For each type of encryption, the file includes two entries, one for the UPN and another for the SPN. Each key is derived from the service account's generated password, so for each encryption type the keys for the UPN and the SPN are identical. In our example, the vastool command generates a keytab file named `SAS.keytab` with contents that look something like this:

```
Type                    Principal                                 Key
aes128-cts-hmac-sha1-96 machineA-SAS@COMPANY.COM                  ca17fd3d8...
aes128-cts-hmac-sha1-96 SAS/machineA.unx.company.com@COMPANY.COM  ca17fd3d8...
aes256-cts-hmac-sha1-96 machineA-SAS@COMPANY.COM                  01562e774...
aes256-cts-hmac-sha1-96 SAS/machineA.unx.company.com@COMPANY.COM  01562e774...
arcfour-hmac-md5        machineA-SAS@COMPANY.COM                  tht8qrg72...
arcfour-hmac-md5        SAS/machineA.unx.company.com@COMPANY.COM  tht8qrg72...
```

Note: *Not all of the encryption types that are listed in a keytab file are necessarily available or used in all contexts.*

- By default, the keytab file is named *service*`.keytab` and is located in `/etc/opt/quest/vas`. You can specify a different location, for example:

```
vastool -u admin service create -k /etc/mypath/SAS.keytab
SAS/
```

For more information, consult the documentation about vastool from Quest.

3. Participating SAS processes on UNIX must be able to read the keytab file. In the standard configuration, those processes run under the SAS Installer (sas) account, so it is that UNIX identity that requires access to the keytab file.

CAUTION: *Anyone who can read a keytab file can use all of the keys that it contains. Make sure that the keytab file is not generally available.*

4. Set the Quest shared library path, based on the host you are working with:

AIX

Add the following code to the `level_env.sh` script in order to set the Quest library path environment variable. Specify locations as appropriate for your environment.

```
SAS_QUEST_PATH="/opt/quest/lib"   # user defines this path for
their AIX platform
    if [ -z "$LIBPATH" ];
    then
      LIBPATH="$SAS_QUEST_PATH"
    else
      LIBPATH="$LIBPATH:$SAS_QUEST_PATH"
    fi
    export LIBPATH
```

HP-UX and HP-UX for the Itanium Processor Family

Add the following code to the `level_env.sh` script in order to set the Quest library path environment variable. Specify locations as appropriate for your environment.

```
SAS_QUEST_PATH="/opt/quest/lib"   # user defines this path for
their HP-UX platform
    if [ -z "$LD_LIBRARY_PATH" ];
    then
      LD_LIBRARY_PATH="$SAS_QUEST_PATH"
    else
      LD_LIBRARY_PATH="$LD_LIBRARY_PATH:$SAS_QUEST_PATH"
    fi
    export LD_LIBRARY_PATH
```

In addition, the path you specified must be added to the file `/etc/dld.sl.conf` to allow for correct dynamic linking of setuid root programs (like sasauth). If the `/etc/dld.sl.conf` file does not exist, it must be created. The file can be readable by all but must only be writeable by root or it will be ignored. See “man dld.so” for more information.

Linux and Linux for x64

Create a file in `/etc/ld.so.conf.d` called `vas.conf`. In this file, add the following line, based on the host you are working with:

- Linux: `/opt/quest/lib`
- Linux for x64: `/opt/quest/lib64`

Note that the added content is based on the default location for installation and may vary.

Run the `/sbin/ldconfig` to recreate the `/etc/ld.so.cache`. This is required for sasauth to perform Kerberos authentication. It runs as root, and shared libraries must be in a trusted path and cannot be specified with LD_LIBRARY_PATH.

Solaris and Solaris for x64

Use the crle command to add the Quest library location to the search path of both the default and trusted search paths:

- Solaris: `/opt/quest/lib/sparcv9`
- Solaris for x64: `/opt/quest/lib/64`

The command would look like this, using Solaris for x64 as an example:

```
crle -64 -c /var/ld/64/ld.config -l
/lib/64:/usr/lib/64:/opt/quest/lib/64 -s
/lib/secure/64:/usr/lib/secure/64:/opt/quest/lib/64
```

After Configuring Your Deployment

1. In `/…/Lev1/level_env.sh` add the following lines. Note that the path may be different for you depending on where you placed the keytab file:

   ```
   KRB5_KTNAME=/etc/opt/quest/vas/SAS.keytab
   export KRB5_KTNAME
   ```

2. Restart the back-end servers to pick up the new environment variable. At this point, the back-end servers should be ready to accept Kerberos connections.

Logins for Users Who Participate in Integrated Windows Authentication

If you choose to configure IWA, make sure that user metadata definitions include logins with properly formatted user IDs. The format of the stored user IDs must match the format in which authenticated user IDs are returned to the target server. Failure to meet this requirement causes the user to have only the generic PUBLIC identity (which, by default, can't even log on to most applications).

In the standard configuration, the appropriate format varies as follows:

- If the target server is on Windows, the authenticated user ID is returned in qualified format, so the stored user ID should be qualified (for example, `WIN\joe` or `fred.smith@company.com`).
- If the target server is on UNIX, the authenticated user ID is returned in short format (it is not qualified), so the stored user ID should not be qualified (for example, `joe` or `fred.smith`).

If you need to align formats, use the SASUSEKERBNAME environment variable. For example, you might use this environment variable in either of the following circumstances:

- The metadata server is on Windows, the workspace server is on UNIX, both are using IWA, and you don't want to store two logins for each user.
- You need to distinguish between two different users, in two different Kerberos realms, who happen to have the same sAMAccountName name (for example, joe@US.COMPANY.COM and joe@EMEA.COMPANY.COM).

For more information, see "Windows User ID Formats" in the *SAS 9.3 Intelligence Platform: Security Administration Guide*, located at `http://support.sas.com/93iwa`.

Using Custom Service Principal Names

In the unusual circumstance where you need to use a SPN that differs from the standard, generated SPN, review the following information.

In a standard configuration on Windows, SAS servers automatically register their SPN as **SAS/***machine* (for example, **SAS/machineA.na.company.com**). Clients can construct the default SPN (because they know the format and machine name), so you don't have to explicitly provide the SPN.

If you need to use a custom SPN on UNIX, the SPN that is used must be listed in the keytab file. In addition to running setspn to set a custom SPN, and adjusting client connection profiles to use that custom SPN, you must generate a new keytab file that includes the new SPN. See step 2 in "Prerequisites for Integrated Windows Authentication on UNIX" above.

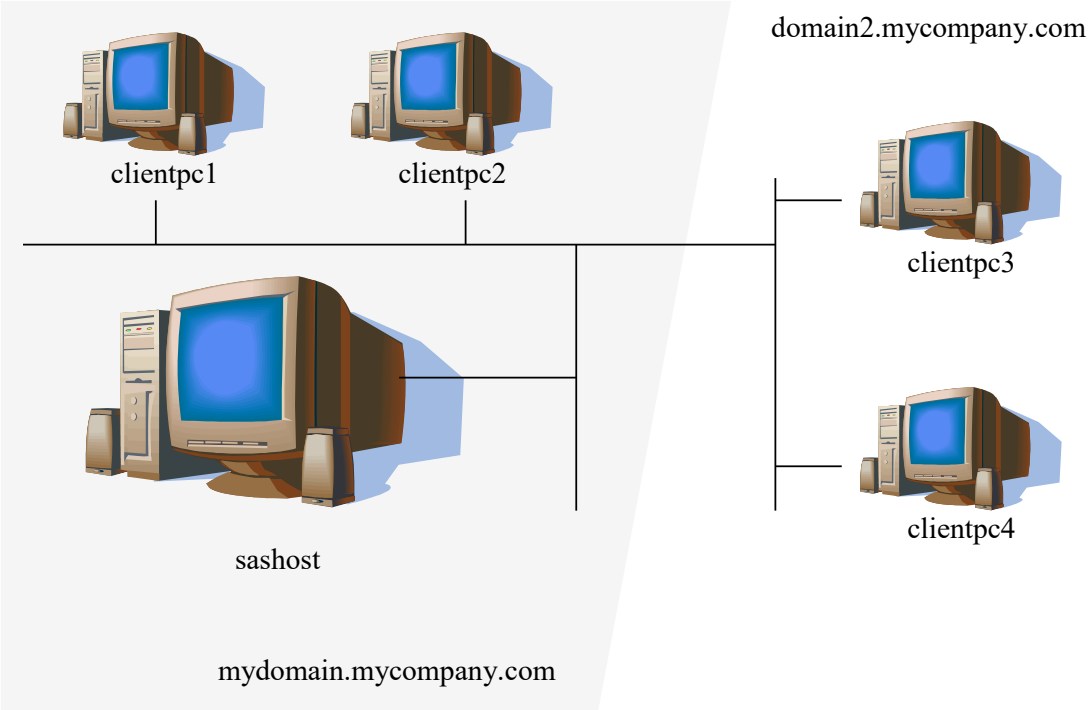
Additional Documentation

For more information about Integrated Windows Authorization, including detailed configuration information about different kinds of servers and recommended security protocols, refer to the "How to Configure Integrated Windows Authentication" topic in the "Authentication Tasks" chapter of the *SAS Intelligence Platform: Security Administration Guide*, located at `http://support.sas.com/93iwa`.

Chapter 6 – Post-Installation Configuration for Remote Browsing

The SAS host may need to be configured appropriately for remote browsing. If one or more SAS desktop clients reside outside the DNS domain of the SAS host, then the host must be configured with a hostname that contains the fully qualified domain name (FQDN) of the host.

For example. SAS is installed on the host `sashost.mycompany.com`, two client computers exist in the same domain (`client1.mycompany.com` and `client2.mycompany.com`), and two other clients exist in another domain (`client3.domain2.mycompany.com` and `client4.domain2.mycompany.com`). This relationship is illustrated below.



If the system sas.mycompany.com is not configured with a hostname that is the FQDN for the system, then client1 and client2 will be able to view HTML content from SAS, but client3 and client4 will not. This is because URLs generated for the SAS host will not include the domain, as in `http://sashost:12345/output.html`.

Since client1 and client2 are in the same domain as SAS itself, their browser will build valid hostnames from their domain, sashost.mycompany.com. But client3 and client4, which are outside the domain of the SAS host, will use their domain names to construct a complete hostname, which results in the invalid name `sashost.domain2.mycompany.com`.

By configuring the SAS host with the system's FQDN, URLs for HTML display are valid from all of clients. From the example, the valid URL for all clients is `http://sashost.mycompany.com:12345/output.html`.

Configuring a Host with a Fully Qualified Domain Name

Note: *Superuser privileges are required to make this change.*

1. Edit `/etc/hosts`.
2. For the IP address of network interfaces for the host, add the FQDN as the first name in the list. For example (using IPv4 addresses):

   ```
   10.4.86.62        sashost
   ```

   becomes

   ```
   10.4.86.62        sashost.mycompany.com sashost
   ```

Chapter 7 – Supporting 64KB pages on AIX Machines

IBM pSeries servers running AIX 5.3 now support 64KB pages as well as 4KB pages. For SAS executables to take advantage of 64KB pages, you should set and export the environment variables using the following commands:

```
$ LDR_CNTRL="DATAPSIZE=64K@TEXTPSIZE=64K@STACKPSIZE=64K@$LDR_CNTRL"
$ export LDR_CNTRL
```

Using 64KB pages rather than 4KB pages for a multi-threaded process's data may reduce the maximum number of threads a process can create due to alignment requirements for stack guard pages. Applications that encounter this limit may disable stack guard pages by setting the environment variable `AIXTHREAD_GUARDPAGES` to 0. (Note that this is really only a problem for 32-bit applications that create many threads, because of the 256M segment address limit in PPC 32-bit mode. Real memory is not allocated for guard pages. This is not a problem for 64-bit programs like SAS 9.3.) Use the following commands to set the `AIXTHREAD_GUARDPAGES` variable correctly (note that this setting is not needed for 64-bit programs).

```
$ AIXTHREAD_GUARDPAGES=0
$ export AIXTHREAD_GUARDPAGES
```

Chapter 8 – Post-Installation Configuration for National Language Support (NLS)

This chapter contains information on post-installation configuration for Asian and European language support.

Important: Before invoking a localized SAS 9.3 Foundation image from a UNIX shell, you must ensure that the UNIX locale environment variable LANG is set appropriately for the language of the SAS version you want to run. The exact values to set will vary depending on your operating system support. To list the locales supported on your operating system, enter the following command:

```
$ locale -a
```

For example, to invoke a Japanese version of SAS 9.3 Foundation in the HP-UX Korn shell environment, enter the following command:

```
$ LANG=ja_JP.SJIS; export LANG
```

For more information on setting locale environment variables, consult the documentation for your operating system.

Introduction

SAS Invocation Scripts

SAS is invoked by Bourne Shell scripts located in the `!SASROOT/bin` directory. A SAS invocation script is created for each language installed. The invocation scripts are named using the language codes of the installed language. For example, `sas_en` invokes the English version of SAS 9.3 Foundation. Below is a list of the valid languages and language codes.

Language	Code
Arabic	AR
Chinese (Simplified)	ZH
Chinese (Traditional)	ZT [EUCTW/BIG5]*
Danish	DA
Dutch	NL
French	FR
German	DE
Hebrew	IW
Hungarian	HU
Italian	IT
Japanese Primary Encoding	JA
Japanese Secondary Encoding	JA [EUC/SJIS]**

Korean	KO
Norwegian	NO
Polish	PL
Portuguese (Brazil)	PB
Portuguese	PT
Russian	RU
Spanish (Castilian)	ES
Swedish	SV
Turkish	TR

* For Chinese Traditional, EUCTW is the primary encoding on Solaris and the secondary encoding on HP-UX. BIG5 is the primary encoding for HP-UX and the secondary encoding for Solaris.

**EUC is the Japanese Secondary Encoding for HP-UX and AIX. SJIS is the Japanese Secondary Encoding for Solaris, and Linux.

SAS Configuration Files

SAS 9.3 Foundation creates a separate configuration file for each language installed (including English). These language-specific configuration files are `!SASROOT/nls/`*`lang`*`/sasv9.cfg` for each respective language. An additional configuration file that is language independent is `!SASROOT/sasv9.cfg`. This master configuration file in `!SASROOT` is used by all languages in addition to the language-specific files in `!SASROOT/nls/`*`lang`*`/`.

Selecting LOCALE during SAS Foundation Deployment

A new dialog was added to the SAS Deployment Wizard (SDW) SAS 9.3 that allows the installer to select the locale to use for SASFoundation. The locale that initially displays in this dialog is the user locale on the UNIX machine where SASFoundation is being installed. If you prefer to use a different locale, you can select a locale from the dialog.

The selected locale is used as the value of the LOCALE system option in the language-specific configuration file that matches the locale. If the selected locale matches a localization installed for the SASFoundation image, the `!SASROOT/sas` symbolic link is set to the SAS invocation script for that localization. Otherwise, the `!SASROOT/sas` symbolic link is set to the appropriate English language script, which is:

`!SASROOT/bin/sas_dbcs` for any language that requires DBCS support

or

`!SASROOT/bin/sas_en` for all other languages.

For example, if the French localization is installed and the French (Canada) [fr_CA] locale is selected, `!SASROOT/sas` is a symbolic link to `!SASROOT/bin/sas_fr`.

Chinese, Japanese, and Korean DBCS Support

This section explains how to specify Asian font catalogs and how to determine the localization used for Chinese locales.

Also, be aware that full-screen products are NOT supported in 9.3 SAS for the following UNIX platforms and languages:

- HP-UX IPF: Japanese, Korean, Simplified Chinese, and Traditional Chinese
- AIX: Korean, Simplified Chinese, and Traditional Chinese

Setting System Fonts with X Resource Files

SAS 9.3 Foundation may not have the correct font settings for your locale by default. To ensure that the correct fonts are defined for the SAS System, you must add them to your X Resource files.

Japanese X Resource template files containing DBCS font settings are located in `!SASROOT/X11/resource_files`, as follows:

- `./Resource_CDE.ja` - for the CDE environment
- `./Resource_LNX.ja` - for Linux
- `./Resource_Sun.ja` - for Solaris
- `./Resource_HP.ja` - for HP-UX
- `./Resource_IBM.ja` - for AIX
- `./Resource_ReflX.ja` - for ReflectionX users

Simplified Chinese X Resource template files containing DBCS font settings are located in `!SASROOT/X11/resource_files`, as follows:

- `./Resource_HP.zh` - for HP-UX
- `./Resource_LNX.zh` - for Linux
- `./Resource_Sun.zh` - for Solaris

Traditional Chinese X Resource template files containing DBCS font settings are located in `!SASROOT/X11/resource_files`, as follows:

- `./Resource_HP.zt` - for HP-UX
- `./Resource_HP.zt.euc` - for HP-UX
- `./Resource_LNX.zt` – for Linux
- `./Resource_Sun.zt` – for Solaris
- `./Resource_Sun.zt.big5` - for Solaris

Korean X Resource template files containing DBCS font settings are located in `!SASROOT/X11/resource_files`, as follows:

- `./Resource_HP.ko` - for HP-UX
- `./Resource_LNX.ko` - for Linux
- `./Resource_Sun.ko` - for Solaris

To apply the X Resources in these template files, copy the appropriate template to one of the following locations, renaming it to `SAS` (in all uppercase):

- `/usr/lib/X11/app-defaults` (on most UNIX systems)
- `/usr/openwin/lib/X11/app-defaults` (on Solaris)
- `$HOME` (your home directory)

For example, on a Solaris system, you would use the following COPY command:

```
$ cp !SASROOT/X11/resource_files/Resource_CDE.ja  /usr/openwin/lib/X11/app-defaults/SAS
```

In the example, `!SASROOT` refers to the root directory of your SAS 9.3 Foundation installation.

For more details, refer to the SAS 9.3 National Language Support (NLS) User's Guide.

Asian Font Catalogs

For SAS 9.3, Simplified and Traditional Chinese have been added to SASHELP.FONTS.

Specifying the Font Catalog in the Configuration File for Traditional Chinese Fonts

When you run a Traditional Chinese localization, the configuration file contains the GFONT definition for the location of the ZT font catalog in the UNIX DBCS directory. However, when you run the English version with LOCALE=ZH_TW, you must either set GFONT in your SAS session or you must modify the DBCS configuration file to define the GFONT definition for the ZT catalog, as follows

```
-set gfontx !SASROOT/nls/zt/font-name
```

In this statement

x represents a value from 0-9

font-name represents the name of the font catalog you want to use.

Specifying the Font Catalog in a SAS Session for Traditional Chinese Fonts

To specify the font catalog in a SAS session, submit the following LIBNAME statement:

```
libname gfontx !SASROOT/nls/zt/font-name
```

In this statement

x represents a value from 0-9

font-name represents the name of the font catalog you want to use.

Chinese Localizations

The installer has the option to install localizations for both Simplified Chinese and Traditional Chinese. Several Chinese-based locales are supported by SAS. In some cases, the localization selected for the locale may not be intuitive. The following table shows which language SAS uses when you select one of the five Chinese locales. Note that the default language may be English.

Locale	Location of `sasv9.cfg` file	Language
Chinese (China) [zh_CN]	`!SASROOT/nls/zh`	Simplified Chinese
Chinese (Hong Kong) [zh_HK]	`!SASROOT/nls/zt`	Traditional Chinese
Chinese (Macau) [zh_MO]	`!SASROOT/nls/dbcs`	English
Chinese (Singapore) [zh_SG]	`!SASROOT/nls/dbcs`	English
Chinese (Taiwan) [zh_TW]	`!SASROOT/nls/zt`	Traditional Chinese

Invoking SAS for Secondary Japanese Encodings

If you have installed the secondary encoding for Japanese you may invoke it as follows:

- If "Secondary Japanese encoding" is installed on HP-UX or AIX, the script `!sasroot/bin/sas_ja.euc` will be created, and it should be used to run the secondary encoding.
- If "Secondary Japanese encoding" is installed on Solaris or Linux, the script `!sasroot/bin/sas_ja.sjis` will be created, and it should be used to run the secondary encoding.

Note: *The SAS deployment tools that create the symbolic link* `!sasroot/bin/sas` *will not point it directly to a script for a secondary encoding. If SAS will mostly be run using either the secondary encoding for either Chinese Traditional or Japanese, you can update* `!sasroot/bin/sas` *so the default SAS command points to the appropriate script.*

Chapter 9 – Configuring SAS Analytics Accelerator for Teradata

This chapter describes registering user-defined function (UDFs) that are included with your SAS Analytics Accelerator software in your Teradata database. The chapter assumes that you have successfully installed SAS Foundation, including SAS/STAT. To leverage Enterprise Miner and ETS UDFs, you also need to install SAS Enterprise Miner and SAS/ETS, respectively.

UDF Installation Step Requires LATIN1

SAS in-Database analytics procedures can be run with all supported encodings. However, the UDF installation step requires `LATIN1` for the session encoding. If the system has a different encoding, it can be temporarily set to `LATIN1` by changing the configuration file as described in *SAS 9.3 National Language Support (NLS): Reference Guide* at this location: `http://support.sas.com/documentation/cdl/en/nlsref/63072/HTML/default/viewer.htm#titlepage.htm`.

Database Permission for Registering the UDFs

Since the SAS Analytics Accelerator UDFs are going to be registered in database SYSLIB, the Teradata database user account you use to install the UDFs as described below must have the following privileges for the SYSLIB database.

- `CREATE FUNCTION`
- `ALTER FUNCTION`
- `EXECUTE FUNCTION`
- `GLOP`
- `GLOP MEMBER`

To obtain the appropriate permissions, contact your database administrator.

Database Requirements and Configuration

To successfully install the UDFs and execute them, the Teradata database must be version 13.00 or higher. In addition, the database must have `DBCEXTENSION` installed to support operations on GLOP sets. Contact your database administrator to ensure `DBCEXTENSION` has been installed prior to taking the next steps.

Registering the UDFs

To register the SAS Analytics Accelerator UDFs in your Teradata database, you should invoke three installation macros called `udftdstt.sas`, `udftdem.sas` and `udftdets.sas` that have been installed under `!SASROOT/stat/sasmacro/` as part of SAS Analytics Accelerator.

Note: *During UDF installation, SAS creates temporary files under the folder referenced by the "work" library. If the absolute path of this folder is very long, UDF installation may fail due to the limit on external file names in Teradata Warehouse. If you encounter this situation, start the SAS session with the work library temporarily assigned to a directory with shorted path, e.g., "`C:\`" and proceed with the installation as described above. After installation is complete, the work*

library can be reassigned to the original folder. For information about the options available when reassigning the work library, refer to SAS documentation.

To run the macros, submit the following commands in the Program Editor from SAS:

```
ods html select none;
ods listing;
%let indconn = server=myserver user=myuserid password=XXXX
database=SYSLIB;
%udftdstt;
%udftdem;
%udftdets;


proc tssql nolibs noerrorstop noprompt="(&credentials.)";
 CREATE GLOP SET syslib.sas_vars;
 call DBCExtension.glop_add('syslib.sas_vars', 'SE', NULL,
'dmdb', 'N', 0, 'Y', 'M', 'E', 0, 256000, 1, '00'XB);
 CREATE GLOP SET syslib.sas_dmvars;
 call DBCExtension.glop_add('syslib.sas_dmvars', 'SE', NULL,
'dmine', 'N', 0, 'Y', 'M', 'E', 0, 256000, 1, '00'XB);
quit;
```

The first two statements in the program allow the registration macros to write and read temporary files without these files being redirected to the default HTML ODS destination of the SAS dms mode. The `INDCONN` macro variable provides credentials to connect to the Teradata machine. You must specify the server, user, password, and database to access the machine on which you have installed the Teradata data warehouse:

- *`myserver`* is the server on which the Teradata warehouse resides.
- *`myuserid`* is a valid user ID for that server that is granted the permissions described above.
- `XXXX` is the password for the user ID you are using.
- Because the SAS Analytics Accelerator UDFs must be registered in the `SYSLIB` database, database must be `SYSLIB`.

The statements executed by `PROC TSSQL` will create the GLOP sets and add the GLOPs used by SAS analytical procedures while executing the UDFs in Teradata database.

Alternative to PROC TSSQL

As an alternative to running `PROC TSSQL`, your database administrator may directly execute the following SQL commands on Teradata through the database client such as BTEQ.

```
CREATE GLOP SET syslib.sas_vars;
call DBCExtension.glop_add('syslib.sas_vars', 'SE', NULL,
      'dmdb', 'N', 0, 'Y', 'M', 'E', 0, 256000, 1, '00'XB);
CREATE GLOP SET syslib.sas_dmvars;
```

```
call DBCExtension.glop_add('syslib.sas_dmvars', 'SE', NULL,
      'dmine', 'N', 0, 'Y', 'M', 'E', 0, 256000, 1, '00'XB);
```

Re-enable the Default HTML ODS Destination

After the UDF registration macros have completed, you can re-enable the default HTML ODS destination by running the following command:

```
ods html select all;
```

Documentation for Using the UDFs

For information about how to use your newly registered UDFs, see the *SAS Analytics Accelerator for Teradata: Guide* at http://support.sas.com/documentation/onlinedoc/analyticsaccel/index.html.

Chapter 10 – Post-Installation Configuration for SAS/ACCESS Software

Before beginning your SAS/ACCESS software configuration, you should determine the following information about your DBMS:

- The version or release of the DBMS client shared libraries installed on your operating system. This is important due to potential incompatibilities between DBMS versions or releases.
- The location of the DBMS client shared libraries. This is important so that SAS/ACCESS software can be loaded at execution time.

Refer to the following sections for detailed DBMS-specific instructions on configuring your environment to interface with your SAS/ACCESS software.

SAS/ACCESS Interface to Aster nCluster

Installing and Configuring the ODBC Driver and Bulk Loader

Before configuring the ODBC driver, the bulk loader should be installed in *SASHOME*/SASFoundation/9.3/ or somewhere that is in the PATH environment variable.

The odbcinst.ini system information file contains the driver definition to connect to your to Aster nCluster server. You must configure the default Aster nCluster driver to use the SAS/ACCESS Interface to Aster nCluster. A sample odbcinst.ini file may be included with the Aster nCluster ODBC Driver. You will have to edit the odbcinst.ini file with a text editor to configure the driver. The general format of the odbcinst.ini file is shown below:

```
[AsterDriver]
Driver=path-to-driver-install/ODBCDriver/libAsterDriver.so
IconvEncoding=UCS-4LE
```

After you configure your driver, you must set the ODBCSYSINI environment variable to the location of your odbcinst.ini:

- For Bourne Shell
  ```
  ODBCSYSINI=path-to-driver-install/Setup
  export ODBCSYSINI
  ```
- For C Shell
  ```
  setenv ODBCSYSINI path-to-driver-install/Setup
  ```

The odbc.ini system information file contains a list of possible data sources to connect to your Aster nCluster servers. Optionally, you can configure at least one data source to use the SAS/ACCESS Interface to Aster nCluster. A sample odbc.ini file may be included with the Aster nCluster ODBC Driver. You will have to edit the odbc.ini file with a text editor to configure the data sources. The general format of the odbc.ini file is shown below:

```
[ODBC Data Sources]
nCluster=AsterDriver


[nCluster]
Driver=AsterDriver
```

```
DATABASE=beehive
SERVER=127.0.0.1
UID=beehive
PWD=beehive
   PORT=2406
```

After you configure your data sources, if `odbc.ini` is not located in the path that the ODBCSYSINI environment variable is set to, you must set the ODBCINI environment variable to the location and name of your `odbc.ini`:

- For Bourne Shell
  ```
  ODBCINI=path-to/odbc.ini
  export ODBCINI
  ```

- For C Shell
  ```
  setenv ODBCINI path-to/odbc.ini
  ```

Finally, you must include the full path to the driver manager shared libraries in the shared library path as shown below so that the driver manager can be loaded dynamically at run time.

Linux for Intel Architecture and Linux for x64	
Bourne Shell	`$ LD_LIBRARY_PATH= path-to-driver-install/Libs:path-to-driver-install/ODBCDriver:$LD_LIBRARY_PATH` `$ export LD_LIBRARY_PATH`
C Shell	`$ setenv LD_LIBRARY_PATH path-to-driver-install/Libs:path-to-driver-install/ODBCDriver:${LD_LIBRARY_PATH}`

SAS/ACCESS Interface to DB2

The SAS/ACCESS Interface to DB2 executable uses shared libraries, referred to in UNIX as shared objects. You must add the location of the shared libraries to one of the system environment variables, and, if necessary, indicate the DB2 version that you have installed at your site. You must also set the `INSTHOME` environment variable to your DB2 home directory before setting the environment variables as shown in the examples.

AIX	
Bourne Shell	`$ LIBPATH=$INSTHOME/lib64:$LIBPATH` `$ export LIBPATH`
C Shell	`$ setenv LIBPATH $INSTHOME/lib64:$LIBPATH`
HP-UX and HP-UX for the Itanium Processor Family Architecture	
Bourne Shell	`$ SHLIB_PATH=$INSTHOME/lib64:$SHLIB_PATH` `$ export SHLIB_PATH`
C Shell	`$ setenv SHLIB_PATH $INSTHOME/lib64:$SHLIB_PATH`
Linux for Intel Architecture	
Bourne Shell	`$ LD_LIBRARY_PATH=$INSTHOME/lib:$LD_LIBRARY_PATH` `$ export LD_LIBRARY_PATH`
C Shell	`$ setenv LD_LIBRARY_PATH $INSTHOME/lib:$LD_LIBRARY_PATH`

Linux for x64, Solaris, and Solaris for x64	
Bourne Shell	`$ LD_LIBRARY_PATH=$INSTHOME/lib64:$LD_LIBRARY_PATH` `$ export LD_LIBRARY_PATH`
C Shell	`$ setenv LD_LIBRARY_PATH $INSTHOME/lib64:$LD_LIBRARY_PATH`

SAS/ACCESS Interface to Greenplum

During the initial installation of SAS/ACCESS Interface to Greenplum, the SAS Deployment Wizard provides a dialog box in which you can specify the location for the required ODBC driver. Use SAS Deployment Manager if you want to update that location after the initial installation. The procedure for updating the location is described in the "Configure SAS/ACCESS Interface to Greenplum" section of *SAS Deployment Wizard and SAS Deployment Manager 9.3: User's Guide*, located at http://support.sas.com/deploywizug93.html.

This location is the ODBCHOME directory, which is used to set up the paths to the shared libraries as well as the `odbc.ini` file below. You must set the ODBCHOME environment variable to your ODBC home directory before setting the ODBCINI and shared library environment variables as shown in the examples below.

The odbc.ini system information file contains a list of possible data sources to connect to your Greenplum servers. You must configure at least one data source in order to use the SAS/ACCESS Interface to Greenplum. Edit the odbc.ini file with a text editor to configure the data sources. The general format of the odbc.ini file is described below:

```
[ODBC Data Sources]
greenplum=SAS ACCESS to Greenplum


[ODBC]
InstallDir=<install_path>
Trace=0
TraceDll=<install_path>/lib/odbctrac.so
TraceFile=odbctrace.out


[greenplum]
Driver=<install_path>/lib/S0gplm<file version>.so
Description=SAS ACCESS to Greenplum
AlternateServers=
ApplicationUsingThreads=1
ConnectionReset=0
ConnectionRetryCount=0
ConnectionRetryDelay=3
Database=<db>
EnableDescribeParam=1
ExtendedColumnMetadata=0
FailoverGranularity=0
FailoverMode=0
FailoverPreconnect=0
FetchRefCursor=1
FetchTSWTZasTimestamp=0
```

```
FetchTWFSasTime=0
HostName=<Greenplum host>
InitializationString=
LoadBalanceTimeout=0
LoadBalancing=0
LoginTimeout=15
LogonID=
MaxPoolSize=100
MinPoolSize=0
Password=
Pooling=0
PortNumber=<Greenplum server port>
QueryTimeout=0
ReportCodepageConversionErrors=0
TransactionErrorBehavior=1
XMLDescribeType=-10
```

Note that the driver version and the <*file version*> specifications describe specific versions of the DataDirect Greenplum driver that is installed with SAS/ACCESS Interface to Greenplum. The driver version in your odbc.ini will already contain the latest version of the DataDirect driver that SAS ships. Also, the file version will contain a two-number version that will describe the version of the actual driver library. You will not need to update these two version specifications in your odbc.ini file.

Replace all occurrences of <*install_path*> in the sample odbc.ini file with the path and directory where you installed the Greenplum ODBC driver. This must be the same directory that is specified by the `ODBCHOME` environment variable, which you set earlier.

You must also replace <*Greenplum host*> with the IP address or hostname of your Greenplum server, replace <*Greenplum server port*> with the port where your Greenplum server is listening (typically 5432), and replace <*db*> with the name of your Greenplum database in your odbc.ini file.

In the above example, `greenplum` is the name of the configured data source name that is used in the DSN= option when assigning a libname to the SAS/ACCESS Interface to Greenplum engine. A sample completed odbc.ini is shown below for reference:

```
[ODBC Data Sources]
Greenplum=SAS ACCESS to Greenplum

[ODBC]
InstallDir=/TECHDBI/odbc/gpdrv
Trace=0
TraceDll=/TECHDBI/odbc/gpdrv/lib/odbctrac.so
TraceFile=/tmp/odbctrace.out

[greenplum]
Driver=/TECHDBI/odbc/gpdrv/lib/S0gplm60.so
Description=SAS ACCESS to Greenplum
AlternateServers=
ApplicationUsingThreads=1
ConnectionReset=0
ConnectionRetryCount=0
ConnectionRetryDelay=3
Database=sample
EnableDescribeParam=1
```

```
ExtendedColumnMetadata=0
FailoverGranularity=0
FailoverMode=0
FailoverPreconnect=0
FetchRefCursor=1
FetchTSWTZasTimestamp=0
FetchTWFSasTime=0
HostName=host-name.domain.com
InitializationString=
LoadBalanceTimeout=0
LoadBalancing=0
LoginTimeout=15
LogonID=
MaxPoolSize=100
MinPoolSize=0
Password=
Pooling=0
PortNumber=5432
QueryTimeout=0
ReportCodepageConversionErrors=0
TransactionErrorBehavior=1
XMLDescribeType=-10
```

If you plan to use DSN-less connections with SAS/ACCESS Interface to Greenplum, you should also modify the sample odbcinst.ini file to replace all occurrences of the <*install_path*> variable. The format of the odbcinst.ini file is shown below:

```
[ODBC Drivers]
SAS ACCESS to Greenplum=Installed

[ODBC Translators]
OEM to ANSI=Installed

[Administrator]
HelpRootDirectory=<install_path>/adminhelp

[ODBC]
#This section must contain values for DSN-less connections
#if no odbc.ini file exists. If an odbc.ini file exists,
#the values from that [ODBC] section are used.

[SAS ACCESS to Greenplum]
Driver=<install_path>/lib/S0gplm27.so
Setup=<install_path>/lib/S0gplm27.so
APILevel=1
ConnectFunctions=YYY
DriverODBCVer=3.52
FileUsage=0
HelpRootDirectory=<install_path>/help
SQLLevel=0
```

After you configure your data sources, you must set the `ODBCINI` environment variable to the location and name of your odbc.ini:

- For Bourne Shell
  ```
  ODBCINI=$ODBCHOME/odbc.ini
  export ODBCINI
  ```

- For C Shell

```
setenv ODBCINI $ODBCHOME/odbc.ini
```

The DataDirect Greenplum ODBC drivers are ODBC API-compliant shared libraries, referred to in UNIX as shared objects. You must include the full path to the shared libraries in the shared library path as shown below so that the ODBC drivers can be loaded dynamically at run time.

AIX	
Bourne Shell	`$ LIBPATH=$ODBCHOME/lib:$LIBPATH` `$ export LIBPATH`
C Shell	`$ setenv LIBPATH $ODBCHOME/lib:${LIBPATH}`
HP-UX for the Itanium Processor Family Architecture	
Bourne Shell	`$ SHLIB_PATH=$ODBCHOME/lib:$SHLIB_PATH` `$ export SHLIB_PATH`
C Shell	`$ setenv SHLIB_PATH $ODBCHOME/lib:${SHLIB_PATH}`

Linux for Intel Architecture and Linux for x64, Solaris, and Solaris for x64	
Bourne Shell	`$ LD_LIBRARY_PATH=$ODBCHOME/lib:$LD_LIBRARY_PATH` `$ export LD_LIBRARY_PATH`
C Shell	`$ setenv LD_LIBRARY_PATH $ODBCHOME/lib:${LD_LIBRARY_PATH}`

Bulkload

SAS/ACCESS can interface with the Greenplum Client Loader interface for loading large volumes of data. To perform bulkloading, the Greenplum Client Loader Package must be present on the system where you install SAS.

SAS recommends using the "gpfdist" protocol for bulkloading. For this protocol you must set the GPLOAD_HOME environment variable to point to the location where the gpfdist utility allocates the files to be loaded onto the Greenplum tables. Further details on how to use the bulkloading feature can be found in the *SAS/ACCESS 9.3 for Relational Databases: Reference* documentation.

SAS/ACCESS Interface to Hadoop

Before you configure SAS/ACCESS Interface to Hadoop, follow the instructions in "Chapter 3 – Post-Installation Configuration for SAS Foundation" in order to set up the directory containing Hadoop JAR files and the `SAS_HADOOP_JAR_PATH` variable.

Run the Hive Service

SAS/ACCESS reads Hadoop data via a JDBC connection to the Hive Service. The Hive Service typically is launched on the Hadoop Namenode. A good practice is to launch the Hive Service as a daemon that kicks off on system restarts, thereby assuring consistent service. For Hadoop administrators unfamiliar with Hive, a simple test before daemon setup is to start the Hive service at an operating system prompt. For example:

```
$ export HIVE_PORT=10000
$ HIVE_HOME/bin/hive --service hiveserver
```

Data Integrity for Data Not in US-ASCII Format

SAS/ACCESS assumes US-ASCII format for Hive STRING columns. To avoid data integrity issues with data in UTF-8 format, take the following actions:

1. If not present, install the HIVE-2137 patch for the `hive-jdbc` JAR file in your `SAS_HADOOP_JAR_PATH` directory.
2. Set the environment variable SAS_HADOOP_TRANSCODING to YES. For example, on Windows, use the following command:

```
set SAS_HADOOP_TRANSCODING=YES
```

If your Hadoop data is not in US-ASCII or UTF-8 format, SAS/ACCESS cannot transparently convert the data for consumption by SAS. You will need to create a customized solution for this issue.

Security Considerations

Read Access Security

SAS/ACCESS supports standard USER= and PASSWORD= security options. SAS/ACCESS propagates the supplied username and password to the JDBC connection string passed to Hive. However, as of Hive 0.7.1, the Hive Service ignores JDBC user IDs and passwords. Permissions are instead those of the Linux user ID that started the Hive Service. The effect is that any SAS user can connect and have read access to all data accessible to the Hive Service. Hive JDBC security is being addressed by the Hadoop community.

Write Access Security

SAS/ACCESS leverages Hadoop HDFS Streaming to create a new Hive table or to append to an existing one. To write data, the SAS user name supplied must be a valid Linux user ID on the Hadoop server with write permission to both the Hadoop HDFS `/tmp` and the Hive warehouse directory. Hadoop HDFS Streaming currently ignores the password. SAS/ACCESS does not restrict the SAS user from specifying a user name that does not match that users SAS machine login ID.

Default Hadoop HDFS Streaming and Hive Ports

The default Hadoop HDFS Streaming port is 8020. The default Hive Service port is 10000. SAS/ACCESS honors these defaults. If you use the defaults, then SAS connection statements do not require additional override options, thus simplifying SAS code. If you override the defaults, you should communicate the appropriate port numbers to SAS users who will access Hadoop.

Successful SAS/ACCESS Connections

SAS code connects to Hive either with a LIBNAME or a PROC SQL CONNECT TO. The LIBNAME outputs information upon a successful connect whereas PROC SQL is silent on a successful connect. In these examples, Hive is listening on default port 10000 on Hadoop Namenode hadoop01.

Sample LIBNAME connect:

```
libname hdplib hadoop server=hadoop01 user=hadoop_usr
password=hadoop_usr_pwd;
NOTE: Libref HDPLIB was successfully assigned as follows:
   Engine:    HADOOP
   Physical Name: jdbc:hive://hadoop01:10000/default
```

Sample PROC SQL connect:

```
proc sql;
connect to hadoop (server=hadoop01 user=hadoop_usr
password=hadoop_usr_pwd);
```

Unsuccessful SAS/ACCESS Connections

SAS failure to connect can have different causes. Error messages will assist in diagnosing the issue.

In this sample failure, Hive is not active on port 10000 on Hadoop Namenode hadoop01.

```
libname hdplib hadoop server=hadoop01 port=10000 user=hadoop_usr
password=hadoop_usr_pwd;
ERROR: java.sql.SQLException: Could not establish connecton to
hadoop01:10000/default:
    java.net.ConnectException: Connection refused: connect
ERROR: Unable to connect to server or to call the Java
Drivermanager.
ERROR: Error trying to establish connection.
ERROR: Error in the LIBNAME statement.
```

In this sample failure, the hive-metastore JAR file is missing from SAS_HADOOP_JAR_PATH.

```
libname hdplib hadoop server=hadoop01 port=10000 user=hadoop_usr
password=hadoop_usr_pwd;
ERROR: java.lang.NoClassDefFoundError:
org/apache/hadoop/hive/metastore/api/MetaException
ERROR: Unable to connect to server or to call the Java
Drivermanager.
ERROR: Error trying to establish connection.
ERROR: Error in the LIBNAME statement.
```

Starting with Hive

If you do not currently run Hive on your Hadoop server, then your Hadoop data likely resides in HDFS files initially invisible to Hive. To make HDFS files (or other formats) visible to Hive, a Hive CREATE TABLE is issued. The following simple scenario demonstrates accessing HDFS files from Hive using the Hive CLI. For more information, perform a web search for "Hive CLI" and locate the appropriate Apache documentation.

1. Assume there are HDFS files `weblog1.txt` and `weblog2.txt` with data lines containing in order, a date field, a text integer field, and a string field. The fields are comma-delimited and lines \n terminated.

   ```
   $ hadoop fs -ls /user/hadoop/web_data
   Found 2 items
   -rw-r--r--  3 hadoop [owner]     [size/date]
   /user/hadoop/web_data/weblog1.txt
   ```

```
-rw-r--r--  3 hadoop [owner]     [size/date]
/user/hadoop/web_data/weblog2.txt
```

On the Hadoop Namenode, begin by terminating the Hive service if it is running. Next, at a Linux prompt, bring up the Hive CLI:

```
$ hive
```

2. At the Hive command prompt, make the weblogs visible to Hive:

```
hive> CREATE EXTERNAL TABLE weblogs (extract_date STRING,
extract_type INT, webdata STRING) ROW FORMAT DELIMITED FIELDS
TERMINATED BY ',' STORED AS TEXTFILE LOCATION
'/user/hadoop/web_data';
```

3. At the Hive command prompt, test that `weblog1.txt` is now accessible to Hive:

```
hive> SELECT * FROM weblogs LIMIT 1;
```

4. If the SELECT works, quit the Hive CLI and start the Hive Service on default port 10000. For example, if your namenode is hadoop_cluster, a test access from SAS would be

```
libname hdplib hadoop server=hadoop_cluster user=hadoop_usr
password=hadoop_usr_pwd;
data work.weblogs;
set hdplib.weblogs(obs=1);
put _all_;
run;
```

This is a complete but intentionally simple scenario intended for new Hive users. It is likely not representative of a mature Hive environment since the default Hive schema is used implicitly and the Hive default Derby metadata store may be in use. Consult Hadoop and Hive documentation to begin to explore Hive in detail. SAS/ACCESS user documentation provides more information on how SAS/ACCESS interacts with Hive.

Proliferation of Hive Logs Files in `/tmp`

Data access through Hive can create log files in Hadoop HDFS `/tmp`. Over time many log files can accumulate. You may want to disable the logging or periodically run a process to delete logs.

SAS/ACCESS Interface to HP Neoview

The HP Neoview ODBC driver is ODBC API-compliant shared libraries, referred to in UNIX as shared objects. You must include the full path to the shared libraries in the shared library path as shown below so that the ODBC drivers can be loaded dynamically at run time. You must also include the full path to any additional system shared libraries that may be required by the HP Neoview ODBC driver.

Note: *The HP Neoview ODBC driver may require additional operating system libraries, libgcc version 3.4.3 or later and libstdc++ version 6.0 or later. Contact HP Neoview for details.*

AIX	

Bourne Shell	`$ LIBPATH=`*`Neoview_ODBC_driver_install_directory`*`/lib:`
	`Additional_system_library_directory``:$LIBPATH`
	`$ export LIBPATH`

C Shell	`$ setenv LIBPATH` *`Neoview_ODBC_driver_install_directory`* `/lib:` *`Additional_system_library_directory`*`:${LIBPATH}`
Linux for Intel Architecture and Solaris	

Bourne Shell	`$ LD_LIBRARY_PATH=`*`Neoview_ODBC_driver_install_directory`*
	`/lib:`*`Additional_system_library_directory`*`:$LD_LIBRARY_PATH`
	`$ export LD_LIBRARY_PATH`

C Shell	`$ setenv LD_LIBRARY_PATH  Neoview_ODBC_driver_install_directory` `/lib:`*`Additional_system_library_directory`*`:${LD_LIBRARY_PATH}`

HP-UX for the Itanium Processor Family Architecture	

Bourne Shell	`$ SHLIB_PATH=`*`Neoview_ODBC_driver_install_directory`*
	`/lib::$SHLIB_PATH`
	`$ export SHLIB_PATH`

C Shell	`$ setenv SHLIB_PATH` *`Neoview_ODBC_driver_install_directory`* `/lib:${SHLIB_PATH}`

Additional Environment Variables for JNI Transporter on HP-UX for the Itanium Processor Family Architecture

The following environment variables are required for customers using the JNI Transporter with SAS/ACCESS Interface to HP Neoview on HP-UX for the Itanium Processor Family Architecture:

```
export JAVA_HOME=/opt/java1.5/jre
export NVTHOME=Transporter-install-directory
export SHLIB_PATH=Neoview-ODBC-driver-install-directory/
   lib:$NVTHOME/lib
export JNVT_SPAWN=Y
```

Note that the `JAVA_HOME` should NOT point to the SAS-installed Java components since SAS uses a 32-bit JVM internally, but the Transporter layer requires a 64-bit JVM. Also, SAS typically specifies a `JAVA_HOME` setting in the `!SASROOT/bin` environment scripts, `sasenv` and `sasenv_local`. Customers wishing to use Transporter should comment out those lines in the `sasenv` and `sasenv_local` scripts so that the shell setting of `JAVA_HOME` takes precedence.

The `JNVT_SPAWN=Y` is a Transporter environment variable that causes Transporter to be launched in a separate process. This is required because SAS needs to use a 32-bit JVM internally, but Transporter requires a 64-bit JVM. Failing to set this variable can lead to errors such as the following:

```
/usr/lib/hpux64/dld.so: Unsatisfied data symbol 'UseSIGUSR2' in load
module '/opt/java6/jre/lib/IA64W/native_threads/libhpi.so'.
/usr/lib/hpux64/dld.so: Unsatisfied data symbol
'doCloseWithReadPending' in load module
'/opt/java6/jre/lib/IA64W/native_threads/libhpi.so'.

HPI shl_load failed: Unresolved external There was an error trying
to initialize the HPI library.

Please check libhpi in your java installation.
```

SAS/ACCESS Interface to Informix

For SAS 9.1 or higher, SAS/ACCESS Interface to Informix software uses an ODBC interface to access Informix.

You may have to edit the `.odbc.ini` file in your home directory with a text editor to configure data sources. Some ODBC driver vendors may allow system administrators to maintain a centralized copy by setting the environment variable `ODBCINI`. Refer to your ODBC driver's vendor documentation for specific information.

The ODBC drivers are ODBC API-compliant shared libraries, referred to in UNIX as shared objects. You must add the location of the shared libraries to one of the system environment variables so that ODBC drivers can be loaded dynamically at run time. You must also set the

InformixDIR environment variable to your Informix home directory before setting the environment variables as shown in the examples.

AIX	
Bourne Shell	`$ LIBPATH = $InformixDIR/lib/cli:$InformixDIR/lib/esql:$LIBPATH` `$ LIBPATH`
C Shell	`$ setenv LIBPATH` `$ InformixDIR/lib/cli:$InformixDIR/lib/esql:$LIBPATH`
HP-UX and HP-UX for the Itanium Processor Family Architecture	
Bourne Shell	`$ SHLIB_PATH=$InformixDIR/lib/cli:$InformixDIR/lib/esql:$SHLIB_PATH` `$ export SHLIB_PATH`
C Shell	`$ setenv SHLIB_PATH` `$ InformixDIR/lib/cli:$InformixDIR/lib/esql:$SHLIB_PATH`
Linux for x64 and Solaris	
Bourne Shell	`$ LD_LIBRARY_PATH=$InformixDIR/lib/cli:$InformixDIR/lib/esql:$LD_LIBRARY_PATH` `$ export LD_LIBRARY_PATH`
C Shell	`$ setenv LD_LIBRARY_PATH $InformixDIR/lib/cli:$InformixDIR/lib/esql:$LD_LIBRARY_PATH`

SAS/ACCESS Interface to Microsoft SQL Server

Before you can use the SAS/ACCESS Interface to Microsoft SQL Server, the following products are required:

- Base SAS software
- SAS/ACCESS Interface to Microsoft SQL Server
- Microsoft SQL Server Version 7.0 or later

SAS/ACCESS Interface to Microsoft SQL Server requires a 64-bit ODBC driver manager and ODBC driver. These ODBC client components (from Progress/Data Direct) are included with SAS/ACCESS Interface to Microsoft SQL Server and must be installed before using the product. These components are unloaded during the SAS/ACCESS Software Configuration phase of the installation. The setup/configuration procedures are described below.

Later versions of these Progress/Data Direct ODBC client components and also installation instructions may be obtained from the SAS Web site at
http://ftp.sas.com/techsup/download/hotfix/datadirect_sqlserver.html

During the initial installation of SAS/ACCESS Interface to Microsoft SQL, the SAS Deployment Wizard provides a dialog in which you can specify the location for the Microsoft SQL Server ODBC drivers that are required. If you wish to update that location, you can do so by using the SAS Deployment Manager. The procedure for updating the location is described in the "Configure SAS/ACCESS Interface to Microsoft SQL" section of *SAS Deployment Wizard and SAS Deployment Manager 9.3: User's Guide*, located at
http://support.sas.com/deploywizug93.html

The directory described above becomes the ODBCHOME directory, which is used to set up the paths to the shared libraries as well as the odbc.ini file below. You must set the ODBCHOME environment variable to your ODBC home directory before setting the ODBCINI and shared library environment variables as shown in the examples below.

The odbc.ini system information file contains a list of possible data sources to connect to your Microsoft SQL Server servers. You must configure at least one data source in order to use the SAS/ACCESS Interface to Microsoft SQL Server. A sample odbc.ini file is located in the ODBCHOME directory as odbc.ini.sample. You will have to edit the odbc.ini file with a text editor to configure the data sources. The general format of the odbc.ini file is shown below:

```
[ODBC Data Sources]
sqlserver=DataDirect driver-version SQL Server Wire Protocol

[sqlserver]
Driver=install-dir/lib/S0msssfile-version.so
Description=DataDirect driver-version SQL Server Wire Protocol
Address=SQLServer-host,SQLServer-server-port
AnsiNPW=Yes
Database=db-name
LogonID=
Password=
QuotedId=yes

[ODBC]
InstallDir=my-install-dir
Trace=0
TraceDll= my-install-dir/lib/odbctrac.so
TraceFile=odbctrace.out
```

Note that the *driver-version* and *file-version* specifications describe specific versions of the DataDirect Microsoft SQL Server driver that is installed with SAS/ACCESS to Microsoft SQL Server. The *driver-version* in your odbc.ini will already contain the latest version of the DataDirect driver that SAS ships. Also, the *file-version* will contain a two-number version that will describe the version of the actual driver library. You will not need to update these two version specifications in your odbc.ini.

You should replace all occurrences of *install-dir* in the sample odbc.ini with the path and directory name you specified during the SAS/ACCESS Software Configuration for MS SQL Server. This is the same directory that is specified by the ODBCHOME environment variable that you set earlier in this section.

You should also replace *SQLServer-host* with the IP address or named server of your SQL Server machine, *SQLServer-server-port* with the port number that your SQL Server is listening on (typically 1433), and *db-name* with the name of your SQL Server database.

In the above example, sqlserver is the name of the configured data source name that is used in the DSN= option when assigning a libname to the SAS/ACCESS to MS SQL Server engine.

A sample completed odbc.ini is shown below for reference:

```
[ODBC Data Sources]
sqlserver= DataDirect 6.1 SQL Server Wire Protocol
```

```
[sqlserver]
Driver=/install/sas/driver/lib/S0msss19.so
Description= DataDirect 6.1 SQL Server Wire Protocol
Address=199.255.255.255,1433
AnsiNPW=Yes
Database=users
LogonID=
Password=
QuotedId=yes
[ODBC]
InstallDir=/install/sas/driver
Trace=0
TraceDll=/install/sas/driver/lib/odbctrac.so
TraceFile=odbctrace.out
```

After you configure your data sources, you must set the ODBCINI environment variable to the location and name of your `odbc.ini`:

- For Bourne Shell
  ```
  ODBCINI=$ODBCHOME/odbc.ini
  export ODBCINI
  ```

- For C Shell
  ```
  setenv ODBCINI $ODBCHOME/odbc.ini
  ```

The DataDirect Microsoft SQL Server ODBC drivers are ODBC API-compliant shared libraries, referred to in UNIX as shared objects. You must include the full path to the shared libraries in the shared library path as shown below so that the ODBC drivers can be loaded dynamically at run time.

AIX	
Bourne Shell	`$ LIBPATH=$ODBCHOME/lib:$LIBPATH` `$ export LIBPATH`
C Shell	`$ setenv LIBPATH  $ODBCHOME/lib:${LIBPATH}`
HP-UX and HP-UX for the Itanium Processor Family Architecture	
Bourne Shell	`$ SHLIB_PATH=$ODBCHOME/lib:$SHLIB_PATH` `$ export SHLIB_PATH`
C Shell	`$ setenv SHLIB_PATH  $ODBCHOME/lib:${SHLIB_PATH}`
Linux for Intel Architecture, Linux for x64, Solaris, and Solaris for x64	
Bourne Shell	`$ LD_LIBRARY_PATH=$ODBCHOME/lib:$LD_LIBRARY_PATH` `$ export LD_LIBRARY_PATH`
C Shell	`$ setenv LD_LIBRARY_PATH  $ODBCHOME/lib:${LD_LIBRARY_PATH}`

SAS/ACCESS Interface to MySQL

During the initial installation of SAS/ACCESS Interface to MySQL, the SAS Deployment Wizard provides a dialog in which you can specify the version of MySQL that you are using. If you wish to update that information, you can do so by using the SAS Deployment Manager. The procedure for updating the version is described in the "Configure SAS/ACCESS Interface to MySQL" section of *SAS Deployment Wizard and SAS Deployment Manager 9.3: User's Guide*, located at http://support.sas.com/deploywizug93.html.

The SAS/ACCESS Interface to MySQL executable uses the MYSQL Version 5.1 shared client libraries, referred to in UNIX as shared objects. You must add the location of the MySQL shared libraries to the shared library path environment variable specific to your operating system. Modify the shared library variable based on the host and shell you are using, according to the table below. The following table assumes the $MYSQL_LIBDIR environment variable names the directory containing the MYSQL Version 5.1 client libraries (for example, the libmysqlclient.so file).

AIX	
Bourne Shell	`$ LIBPATH=$MYSQL_LIBDIR:$LIBPATH` `$ export LIBPATH`
C Shell	`$ setenv LIBPATH $MYSQL_LIBDIR:$LIBPATH`
HP-UX and HP-UX for the Itanium Processor Family Architecture	
Bourne Shell	`$ SHLIB_PATH=$MYSQL_LIBDIR:$SHLIB_PATH` `$ export SHLIB_PATH`
C Shell	`$ setenv SHLIB_PATH $MYSQL_LIBDIR:$SHLIB_PATH`
Linux for Intel Architecture and Linux for x64	
Bourne Shell	`$ LD_LIBRARY_PATH=$MYSQL_LIBDIR:$LD_LIBRARY_PATH` `$ export LD_LIBRARY_PATH`
C Shell	`$ setenv LD_LIBRARY_PATH $MYSQL_LIBDIR:$LD_LIBRARY_PATH`
Solaris and Solaris for x64	
Bourne Shell	`$ LD_LIBRARY_PATH=$MYSQL_LIBDIR:$LD_LIBRARY_PATH` `$ export LD_LIBRARY_PATH`
C Shell	`$ setenv LD_LIBRARY_PATH $MYSQL_LIBDIR:$LD_LIBRARY_PATH`

If you see the following error message, double-check your library path environment variable.

```
ERROR: The SAS/ACCESS Interface to MYSQL cannot be loaded. The
libmysqlclient code appendage could not be loaded.
ERROR: Error in the LIBNAME statement.
```

SAS/ACCESS Interface to Netezza

During the initial installation of SAS/ACCESS Interface to Netezza, the SAS Deployment Wizard provides a dialog in which you can specify the version of Netezza that you are using. If you wish to update that information, you can do so by using the SAS Deployment Manager. The procedure for updating the version is described in the "Configure SAS/ACCESS Interface to Netezza" section of *SAS Deployment Wizard and SAS Deployment Manager 9.3: User's Guide*, located at `http://support.sas.com/deploywizug93.html`.

The Netezza ODBC drivers are ODBC API-compliant shared libraries, referred to in UNIX as shared objects. You must include the full path to the shared libraries in the shared library path as shown below so that the ODBC drivers can be loaded dynamically at run time.

AIX	
Bourne Shell	`$ LIBPATH=$ODBCHOME/lib64:$LIBPATH` `$ export LIBPATH`

C Shell	`$ setenv LIBPATH  $ODBCHOME/lib64:${LIBPATH}`
HP-UX and HP-UX for the Itanium Processor Family Architecture	
Bourne Shell	`$ SHLIB_PATH=$ODBCHOME/lib64:$SHLIB_PATH` `$ export SHLIB_PATH`
C Shell	`$ setenv SHLIB_PATH  $ODBCHOME/lib64:${SHLIB_PATH}`
Linux for Intel Architecture	
Bourne Shell	`$ LD_LIBRARY_PATH=$ODBCHOME/lib:$LD_LIBRARY_PATH` `$ export LD_LIBRARY_PATH`
C Shell	`$ setenv LD_LIBRARY_PATH  $ODBCHOME/lib:${LD_LIBRARY_PATH}`
Linux for x64 and Solaris	
Bourne Shell	`$ LD_LIBRARY_PATH=$ODBCHOME/lib64:$LD_LIBRARY_PATH` `$ export LD_LIBRARY_PATH`
C Shell	`$ setenv LD_LIBRARY_PATH  $ODBCHOME/lib64:${LD_LIBRARY_PATH}`

SAS/ACCESS Interface to ODBC

You may have to edit the `.odbc.ini` file in your home directory with a text editor to configure data sources. Some ODBC driver vendors may allow system administrators to maintain a centralized copy by setting the environment variable `ODBCINI`. Refer to the vendor's documentation for your ODBC driver to find more specific information.

The ODBC drivers are ODBC API-compliant shared libraries, referred to in UNIX as shared objects. You must add the location of the shared libraries to one of the system environment variables so that ODBC drivers can be loaded dynamically at run time. You must also set the `ODBCHOME` environment variable to your ODBC home directory before setting the environment variables as shown in the examples.

AIX	
Bourne Shell	`$ LIBPATH=$ODBCHOME/lib:$LIBPATH` `$ export LIBPATH`
C Shell	`$ setenv LIBPATH  $ODBCHOME/lib:${LIBPATH}`
HP-UX and HP-UX for the Itanium Processor Family Architecture	
Bourne Shell	`$ SHLIB_PATH=$ODBCHOME/lib:$SHLIB_PATH` `$ export SHLIB_PATH`
C Shell	`$ setenv SHLIB_PATH  $ODBCHOME/lib:${SHLIB_PATH}`
Linux for Intel Architecture and Linux for x64	
Bourne Shell	`$ LD_LIBRARY_PATH=$ODBCHOME/lib:$LD_LIBRARY_PATH` `$ export LD_LIBRARY_PATH`
C Shell	`$ setenv LD_LIBRARY_PATH $ODBCHOME/lib:$LD_LIBRARY_PATH`

Solaris and Solaris for x64	
Bourne Shell	`$ LD_LIBRARY_PATH=$ODBCHOME/lib:$LD_LIBRARY_PATH` `$ export LD_LIBRARY_PATH`
C Shell	`$ setenv LD_LIBRARY_PATH  $ODBCHOME/lib:${LD_LIBRARY_PATH}`

SAS/ACCESS Interface to Oracle

During the initial installation of SAS/ACCESS Interface to Oracle, the SAS Deployment Wizard provides a dialog in which you can specify the version of Oracle that you are using. If you wish to update that information, you can do so by using the SAS Deployment Manager. The procedure for updating the version is described in the "Configure SAS/ACCESS Interface to Oracle" section of *SAS Deployment Wizard and SAS Deployment Manager 9.3: User's Guide*, located at `http://support.sas.com/deploywizug93.html`.

In order to use SAS/ACCESS Interface to Oracle software, you must set the `ORACLE_HOME` environment variable. In addition, you must make sure that the shared library path variable (the name of this variable is operating system dependent) points to where the Oracle shared libraries are located. This is required since the SAS/ACCESS Interface to Oracle executable uses Oracle shared libraries and needs to know where they are located at your site.

The following are examples for the various operating systems:

AIX	
Bourne Shell	`$ LIBPATH=$ORACLE_HOME/lib:$LIBPATH` `$ export LIBPATH`
C Shell	`$ setenv LIBPATH=$ORACLE_HOME/lib:$LIBPATH`
HP-UX and HP-UX for the Itanium Processor Family Architecture	
Bourne Shell	`$ SHLIB_PATH=$ORACLE_HOME/lib:$SHLIB_PATH` `$ export SHLIB_PATH`
C Shell	`$ setenv SHLIB_PATH $ORACLE_HOME/lib:$SHLIB_PATH`
Linux for Intel Architecture, Linux for Itanium-based Systems, Solaris, and Solaris for x64	
Bourne Shell	`$ LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH` `$ export LD_LIBRARY_PATH`
C Shell	`$ setenv LD_LIBRARY_PATH $ORACLE_HOME/lib:$LD_LIBRARY_PATH`

If the environment variables are not set correctly, you will error messages similar to those below when connecting to Oracle from SAS.

```
ERROR: Could not load /app/SAS/SASFoundation/9.3/sasexe/sasora (86
images loaded)
```

```
ERROR: Could not load module /app/SAS/SASFoundation/9.3/sasexe/sasora.
Dependent module libclntsh.a(shr.o) could not be loaded.Could not load
module libclntsh.a(shr.o).System error: No such file or directoryCould
not load module /app/SAS/SASFoundation/9.2/sasexe/sasora. Dependent
module /app/SAS/SASFoundation/9.2/sasexe/sasora could not be loaded.
```

```
ERROR: The SAS/ACCESS Interface to ORACLE cannot be loaded.
ERROR: Image SASORA found but not loadable.
ERROR: Error in the LIBNAME statement.
```

SAS/ACCESS Interface to R/3

SAS/ACCESS Interface to SAP R/3 software requires extensive post-installation configuration before it can be used. For detailed information, refer to the *Configuration Instructions for SAS/ACCESS 4.4 Interface to R/3* on Install Center http://support.sas.com/documentation/installcenter/en/ikr3cg/64225/PDF/default/config.pdf

SAS/ACCESS Interface to Sybase

During the initial installation of SAS/ACCESS Interface to Sybase, the SAS Deployment Wizard provides a dialog in which you can specify the version of Sybase that you are using. If you wish to update that information, you can do so by using the SAS Deployment Manager. The procedure for updating the version is described in the "Configure SAS/ACCESS Interface to Sybase" section of *SAS Deployment Wizard and SAS Deployment Manager 9.3: User's Guide*, located at http://support.sas.com/deploywizug93.html.

For users of Sybase Open Client 15, in order to correctly copy the Sybase libraries for use with SAS/ACCESS Interface to Sybase, you must have read/write authority for `$SYBASE/OCS-15_0/lib` and `$SYBASE/OCS-15_0/devlib` in order to run `$SYBASE/OCS-15_0/scripts/lnsyblib`. Instructions for copying the Sybase libraries are in the header comments in the `lnsyblib` file.

Installing Sybase Procedure

In SAS 9.3, the administrator or user must install two Sybase-stored procedures on the target Sybase server. Two files have been included in the `!SASROOT/misc/dbi` directory to assist in the installation:

- `sas-spcp.txt` is a text file containing instructions on how to do the installation.
- `sas-spdf.txt` is the first of two actual stored procedure scripts for CTLIB 12.5x users
- `sas-spdf_15.txt` is the first of two actual stored procedure scripts for CTLIB 15 users
- `sassp2df.txt` is the second of two stored procedure scripts for CTLIB 12.5x users
- `sassp2df_15.txt` is the second of two stored procedure script for CTLIB 15 users.

The process utilizes two Sybase facilities, `defncopy` and `isql`.

Adding Shared Libraries

Finally, the SAS/ACCESS Interface to Sybase executable uses shared libraries, referred to in UNIX as shared objects. You must add the location of the shared libraries to one of the system environment variables and, if necessary, indicate the Sybase version that you have installed at your site. You must also set the `Sybase` environment variable to your Sybase home directory before setting the environment variables as shown in the examples.

AIX	
Bourne Shell	`$ LIBPATH=$SYBASE/lib:$LIBPATH` `$ export LIBPATH`
C Shell	`$ setenv LIBPATH $SYBASE/lib:$LIBPATH`
HP-UX and HP-UX for the Itanium Processor Family	
Bourne Shell	`$ SHLIB_PATH=$SYBASE/lib:/lib:$SHLIB_PATH` `$ export SHLIB_PATH`
C Shell	`$ setenv SHLIB_PATH $SYBASE/lib:/lib:$SHLIB_PATH`
Linux for Intel Architecture, Linux for x64, Solaris, and Solaris for x64	
Bourne Shell	`$ LD_LIBRARY_PATH=$SYBASE/lib:/lib:$LD_LIBRARY_PATH` `$ export LD_LIBRARY_PATH`
C Shell	`$ setenv LD_LIBRARY_PATH $SYBASE/lib:/lib:$LD_LIBRARY_PATH`

SAS/ACCESS Interface to Sybase IQ

You must first install the Sybase IQ client software available from Sybase. After your installation is complete, you will need to run a script that will set up your environment, including the path to the shared library. This script is located in the installation directory for your Sybase IQ client software. If you are using Sybase IQ 12.7, the script is named `ASIQ-12_7.sh` (or `.csh`). If you are using Sybase IQ 15 or later, the script is named after the version. For example, the script for Sybase IQ 15 is named `IQ-15_0.sh`.

SAS/ACCESS Interface to Teradata

Access to Shared Libraries

The SAS/ACCESS Interface to Teradata executable uses shared libraries, referred to in UNIX as shared objects. These shared objects typically reside in `/usr/lib`. You must add the location of the shared libraries to one of the system environment variables.

AIX	
Bourne Shell	`$ LIBPATH=`*`TERADATA-CLIENT-LOCATION`*`:$LIBPATH` `$ export LIBPATH`
C Shell	`$ setenv LIBPATH `*`TERADATA-CLIENT-LOCATION`*`:$LIBPATH`
HP-UX	
Bourne Shell	`$ SHLIB_PATH=`*`TERADATA-CLIENT-LOCATION`*`:$SHLIB_PATH` `$ export SHLIB_PATH`
C Shell	`$ setenv SHLIB_PATH `*`TERADATA-CLIENT-LOCATION`*`:$SHLIB_PATH`

HP-UX for the Itanium Processor Family	

Bourne Shell	`$ SHLIB_PATH=`*`TERADATA-CLIENT-LOCATION`*`:$SHLIB_PATH`
	`$ export SHLIB_PATH`
	`$ LD_PRELOAD=/usr/lib/hpux64/libpthread.so.1`
	`$ export LD_PRELOAD`

C Shell	`$ setenv SHLIB_PATH `*`TERADATA-CLIENT-LOCATION`*`:$SHLIB_PATH` `$ setenv LD_PRELOAD /usr/lib/hpux64/libpthread.so.1`
Linux for Intel Architecture, Linux for x64, Solaris, and Solaris for x64	
Bourne Shell	`$ LD_LIBRARY_PATH=`*`TERADATA-CLIENT-LOCATION`*`:$LD_LIBRARY_PATH` `$ export LD_LIBRARY_PATH`
C Shell	`$ setenv LD_LIBRARY_PATH `*`TERADATA-CLIENT-LOCATION`*`:$LD_LIBRARY_PATH`

TTU 8.2 and HP-UX

HP-UX users with TTU 8.2 must create two symbolic links from the /usr/lib/pa20_64 directory with the following commands:

```
$ ln -s /usr/lib/pa20_64/libicudatatd.sl libicudatatd.sl.34
$ ln -s /usr/lib/pa20_64/libicuuctd.sl libicuuctd.sl.34
```

FastExporting

For optimal reads of large tables, SAS/ACCESS can perform FastExporting. To perform FastExporting, the Teradata FastExport Utility must be present on the system where you install SAS.

As needed, modify your library path environment variable to include the directory containing `sasaxsm.sl` (HP-UX) or `sasaxsm.so` (Linux, Solaris, and AIX). These shared objects are delivered in the `$SASROOT/sasexe` directory. You may copy these modules where you wish, but ensure that the directory you copy them into is in the appropriate shared library path environment variable.

On Solaris and Linux, the library path variable is LD_LIBRARY_PATH. On HP-UX, it is SHLIB_PATH. On AIX, it is LIBPATH. Also, make sure that the Teradata FastExport utility, `fexp`, has its directory included in the PATH environment variable. This utility is usually installed in the `/usr/bin` directory.

The FastExport Utility is not required; SAS/ACCESS reads large tables quite efficiently without it. For further information, see the DBSLICEPARM option in your SAS/ACCESS to Teradata documentation. Contact Teradata if you want to obtain the Teradata FastExport Utility.

MultiLoad

SAS/ACCESS can interface with MultiLoad for loading large volumes of data. To perform MultiLoading, the Teradata MultiLoad Utility must be present on the system where you install SAS.

As needed, modify your library path environment variable to include the directory containing the shared objects `sasmlam.sl` and `sasmlne.sl` (HP-UX) or `sasmlam.so` and `sasmlne.so` (Linux, Solaris, HP-UX for the Itanium Processor Family and AIX). These shared objects are delivered in the `$SASROOT/sasexe` directory. You may copy these modules where you wish, but ensure that the directory you copy them into is in the appropriate shared library path

environment variable. On Solaris and Linux, the library path variable is LD_LIBRARY_PATH. On HP-UX and HP-UX for the Itanium Processor Family, it is SHLIB_PATH. On AIX, it is LIBPATH. Also, make sure that the Teradata MultiLoad utility, `mload`, has its directory included in the PATH environment variable. This utility is usually installed in the `/usr/bin` directory.

The MultiLoad Utility is not required; SAS/ACCESS provides other options for loading tables. For further information, see the MULTISTMT option in your SAS/ACCESS Interface to Teradata documentation. Contact Teradata if you want to obtain the Teradata MultiLoad Utility.

Teradata Parallel Transporter

SAS/ACCESS supports the Teradata parallel transporter API for loading data using Multiload, Fastload, and multi-statement inserts. The API also supports reading data using FastExport.

Note: The Teradata Parallel Transporter API is not required; SAS/ACCESS provides other options for loading and reading data.

If you plan to use the Teradata parallel transporter API, the following two requirements must be met:

1. The API must be installed on the system where SAS is installed.
2. The path system variable must include the location of the Teradata Parallel transporter API libraries (specifically the location of libtelapi.*). It may be necessary to set other environment variables depending on the type of UNIX environment. Some of these variables may have already been set correctly when Teradata parallel transporter was installed.

   AIX:

   ```
   LIBPATH=TPT-API-LIBRARY-LOCATION:$LIBPATH
   NLSPATH=TPT-API-MESSAGE-CATALOG-LOCATION
   LC__FASTMSG=false             // Note: There are two underscores
   ```

   HP-UX and HP-UX for the Itanium Processor Family:

   ```
   SHLIB_PATH=TPT-API-LIBRARY-LOCATION:$SHLIB_PATH
   NLSPATH=TPT-API-MESSAGE-CATALOG-LOCATION
   ```

   Linux for Intel Architecture, Linux for x64, and Solaris for x64:

   ```
   LD_LIBRARY_PATH=TPT-API-LIBRARY-LOCATION:$LD_LIBRARY_PATH
   NLSPATH=TPT-API-MESSAGE-CATALOG-LOCATION
   ```

Configuring and Administering SAS In-Database Products

Deploying the SAS In-Database products requires detailed configuration and administration steps following the initial installation. Follow the steps in your Software Order E-mail and QuickStart Guide to perform the initial deployment. Then refer to the "Administrator's Guide" chapter of the *SAS In-Database Products: Administrator's Guide* for your particular database.

The "Administrator's Guide" chapter contains instructions on how to install and configure the in-database deployment package for your particular database. When you have completed the instructions described there, your software will be ready for use.

The *SAS In-Database Products: Administrator's Guide* is located at
http://support.sas.com/documentation/onlinedoc/indbtech/index.html.

Chapter 11 – Post-Installation Configuration for SAS/ASSIST Software

This chapter describes how to add a master profile to SAS/ASSIST software. You can use a master profile to override the default SAS settings. This allows you to provide a customized setup for SAS/ASSIST software. With the master profile, you can control the profile options of all SAS/ASSIST users from one central place. For information on the profile options, refer to the *SAS/ASSIST Software System Administrator's Guide.*

Adding a Master Profile

Complete the following steps to add a master profile to SAS/ASSIST software.

1. Specify the location of the master profile by creating a new directory to which all users of SAS/ASSIST software will have Read access.

   All users with write access to this directory will automatically have write access to the master profile in SAS/ASSIST software. Select a name that conforms to the naming conventions of your installation. The name of this new directory must be stored in an entry in the `SASHELP` library. This requires that you have write access to the `SASHELP` library.

   On line 1 of the `Program Editor` window of the SAS Display Manager System, type the physical pathname of the master profile directory. Execute the `Save` command to store this pathname in the `SASHELP.QASSIST` catalog. Save it as `SASHELP.QASSIST.PARMS.SOURCE`. The location of the master profile will now be known by SAS/ASSIST software.

2. Create the master profile.

   The first time SAS/ASSIST software is started, a master profile is created if `SASHELP.QASSIST.PARMS.SOURCE` contains the name of an existing physical pathname, and the person who starts SAS/ASSIST software has write access to this physical pathname.

3. Customize the master profile by starting SAS/ASSIST software and selecting:

   ```
   Setup, then
   Profiles, and then
   Master/group ...
   ```

   If you have write access to the SAS library containing the master profile, you can specify default values. New users will use these default values when they start SAS/ASSIST software.

   Note: *If you restrict values by typing* `R` *in* `Status`*, users will not be allowed to change the values you define.*

   You can run SAS/ASSIST software in two different styles - Workplace or Block Menu. The Block Menu can be New style or Old style. You can control this using the profile options below.

Run workplace:

```
SAS/Assist style:          Workplace
```

Run block menu new style:

```
SAS/Assist style:          Block Menu
Save selections on end:    Yes
Menu Style:                New
```

Run old style:

```
SAS/Assist style:          Block Menu
Save selections on end:    Yes
Menu Style:                Old
```

By setting the default values in the master profile, you can control if users should use the New or Old style of SAS/ASSIST software. In addition, there are many other profile options. For more information on these options, refer to the *SAS/ASSIST Software System Administrator's Guide*.

4. Create group profiles.

   From the master profile, it is possible to create group profiles to allow groups of users to have different setups. The master profile controls group profiles and user profiles when a user is not a member of any group. All users are indirectly controlled by the master profile when option values are set to a restricted status.

   Select `Setup...Master/Group`

   then `Tools...Create Group Profile`.

   To add users to a group profile, select `Tools...Update User Group`.

   By default, the user ID is found in the macro variable `&SYSJOBID`. This value is set in the option `Userid` in the master profile (option type `System Administration`). Change the value if your site uses another variable to keep the user ID. If the value name starts with `&`, it is a macro variable; otherwise, it is an environment variable, which is set before the start of SAS 9.3.

Chapter 12 – Post-Installation Configuration for SAS/CONNECT Software

TCP/IP is the access method supported for UNIX environments and their derivatives. Refer to the publication *Communications Access Methods for SAS/CONNECT and SAS/SHARE Software* for information on the access methods supported by other systems. This document can be found at http://support.sas.com/documentation/onlinedoc/connect/index.html.

Storing and Locating SAS/CONNECT Script Files

SAS/CONNECT software ships several sample script files that are used to establish a connection to a remote SAS session. The `SASSCRIPT` configuration option points to the location of the SAS/CONNECT script files. The `SASSCRIPT` option is used by SAS/ASSIST software and can be used by user-written SCL applications.

The script files are installed into the `!SASROOT/misc/connect` directory by default. The following line has been included in the `sasv9.cfg` file in order to define the default script file location:

```
-SASSCRIPT !SASROOT/misc/connect
```

If you want to move the script files to another directory, you must edit the `sasv9.cfg` file and update the `SASSCRIPT` option with the new directory location.

Configuring the SAS UNIX Spawner Program

The SAS UNIX Spawner is stored in the `!SASROOT/utilities/bin` directory and can be executed manually from the `!SASROOT/utilities/bin` directory at any time. For complete documentation on the UNIX spawner and the supported options, see *Communications Access Methods for SAS/CONNECT 9.3 and SAS/SHARE 9.3*.

Chapter 13 – Post-Installation Configuration for SAS/GRAPH Software

Loading SAS Fonts to Your X Display Server

Many SAS/GRAPH procedures and devices now support ODS styles in all destinations, including the LISTING destination. By default, all colors, fonts, symbols, and graph sizes are derived from the current style. The default fonts in these styles are the TrueType fonts provided by SAS. Devices that use FreeType rendering are able to find these fonts by default and render them in an environment without a DISPLAY set or valid Xdisplay available. For devices like XCOLOR that use host-rendering, the fonts must be registered with the display in order for them to work. You may override the default font setting by using the FTEXT option on the GOPTIONS statement or by creating a modified style sheet. However, SAS recommends you make the TrueType fonts available to the display device to take advantage of their benefits.

Refer to your vendor user documentation for your X display server for the instructions for making the SAS fonts available to it. SAS's fonts are located at `$SASROOT/misc/fonts`.

Making System Fonts Available to SAS

One of the main advantages of using FreeType rendering is that TrueType and other hardware fonts that produce high quality text are available in an environment without a DISPLAY set. The graphics devices that use FreeType rendering will only recognize fonts that have been registered in SAS.

If you wish to register additional fonts to SAS, including system or display fonts, use the FONTREG procedure to update the SAS registry to include these fonts. For full information about the use and syntax of the FONTREG procedure, see the appropriate chapter in the *Base SAS 9.3 Procedures Guide*, available from http://support.sas.com.

Chapter 14 – Post-Installation Configuration for SAS/IntrNet Software

This chapter has information for your SAS/IntrNet installation. It will help you install, configure, and test your SAS/IntrNet components.

The procedures for installing SAS software using the SAS Deployment Wizard are described in other documentation and not available from this chapter. Furthermore, the installation of your Web server is your responsibility and not described in SAS documentation.

When the SAS/IntrNet software has been installed, configured and tested using the procedures described in this chapter, review the latest version of the SAS/IntrNet product documentation online at `http://support.sas.com/documentation/onlinedoc/IntrNet/index.html`. The "What's New" page at this Web site lists any recent changes to the product or documentation.

Overview

All SAS/IntrNet installations are made up of two components:

1. The SAS/IntrNet server (also referred to as the Application Server). This is where SAS Foundation is installed.
2. CGI Tools (also referred to as the Broker). This is where the `broker.cfg` file and its supporting files are installed.

When you install SAS/IntrNet, choose between two installation configurations:

Type A - The SAS/IntrNet server and CGI Tools components are both installed on the same system machine. The Web server **must** be installed before starting the SAS installation.

Type B -The SAS/IntrNet server component is installed on one system machine and the CGI Tools component is installed on a different system machine. The Web server **must** be installed on the CGI Tools system machine prior to installing CGI Tools.

Type A and Type B require different installation steps:

Type A Installation Steps
Confirm that the Web server software (IIS, Apache etc.) is on the same system machine as your SAS/IntrNet software.
Install your SAS products. Check "CGI Tools for the Web Server" in the "Select Products to Install" dialog.

Type B Installation Steps
Confirm that the Web server software (IIS, Apache, etc.) is on the system machine where you will install CGI Tools.
On your application server system machine, start your SAS installation. Uncheck "CGI Tools for the Web Server" in the "Select Products to Install" dialog.
On your Web server system machine, start your SAS installation. Uncheck all products except "CGI Tools for the Web Server" in the "Select Products to Install" dialog. You can optionally check the IntrNet Monitor or Connect Drivers

Test the Broker
Configure a Socket Service
Start the Socket Service
Test the Socket Service

Test the Broker
Configure a Socket Service
Start the Socket Service
Test the Socket Service

The steps are described more thoroughly in the sections that follow.

Installing and Configuring SAS/IntrNet Software

Install Your Web Server Software

Refer to your Web server's documentation for its installation procedures.

Install Your SAS Software

Refer to your *QuickStart Guide* for a description of how to start your SAS software installation.

If you are performing a Type A installation (as described in the "Overview" above), confirm that your Web server software is installed before starting your SAS software installation. Check "CGI Tools for the Web Server" in the **Select Products to Install** dialog.

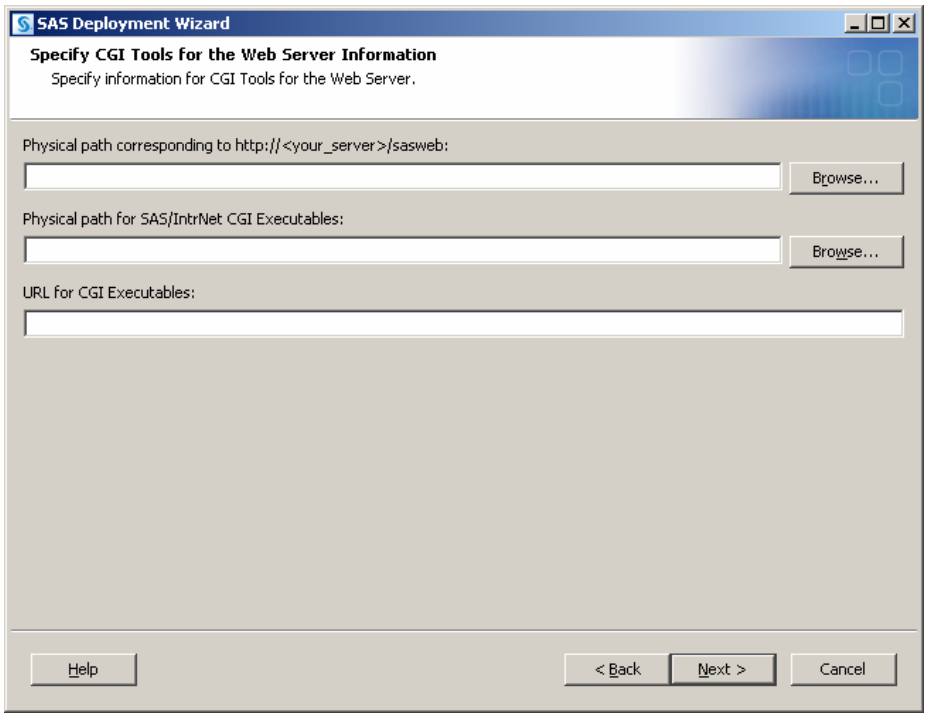
If you are performing a Type B installation (as described in the "Overview" above), do **both** of the following:

- Install the SAS software on the SAS System side, unchecking "CGI Tools for the Web Server" and "SAS/GRAPH Java Applets" in the **Select Products to Install** dialog.
- Start the SAS software install on the Web server and check "CGI Tools for the Web Server" and "SAS/GRAPH Java Applets" in the **Select Products to Install** dialog. SAS/IntrNet Monitor and SAS/CONNECT Driver for Java are optional selections. Uncheck everything else.

CGI Tools Installation Dialogs

The following screens appear for CGI Tools for the Web Server for all installations. Click **Help** on any dialog for information about the fields.

Customary entries are documented following each screen shown below. Customize the entries according to your environment.



The following are examples of common entries for popular Web servers. Customize your entries according to your own web Server environment. These fields will tell SAS where your Web server software is located.

Physical path corresponding to http://*your-server*/sasweb:

IIS:	`C:\Inetpub\wwwroot\sasweb`
Apache (Windows):	`C:\program files\Apache Software Foundation\Apache2.2\ htdocs\sasweb`
Apache (UNIX):	`/usr/local/apache2/htdocs/sasweb`

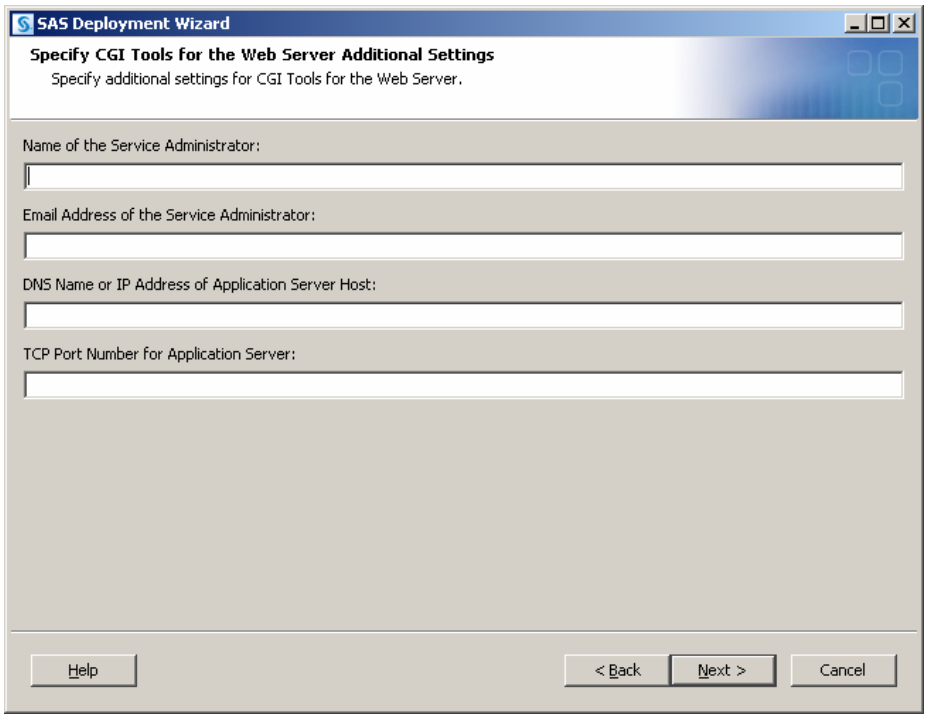
Physical path for SAS/IntrNet CGI Executables:

IIS:	`C:\Inetpub\scripts`
Apache (Windows):	`C:\program files\Apache Software Foundation\Apache2.2 \cgi-bin`
Apache (UNIX):	`/usr/local/apache2/cgi-bin`

URL for CGI Executables:

IIS:	`http://`*`web-servername`*`/scripts`
	Example: `http://abcserver.comp.com/scripts`
Apache (Windows):	`http://`*`web-servername`*`/cgi-bin`
	Example: `http://abcserver.comp.com/cgi-bin`
Apache (UNIX):	`http://`*`web-servername`*`/cgi-bin`

Example: `http://abcserver.comp.com/cgi-bin`



Note that your entries for this dialog are added to the `broker.cfg` file. The `broker.cfg` file is a text file that can be edited after the installation is complete.

Name of the Service Administrator:

(optional) Enter the name of the administrator (for example, John Doe).

Email Address of the Service Administrator:

(optional) Enter the e-mail address of the administrator (for example, NetAdmin@comp.com).

DNS Name or IP Address of Application Server Host:

Enter the DNS name or IP address of the application server host where SAS Foundation is located.

TCP Port Number for Application Server:

The customary default port number is *5001*, but you can use any valid available port on your system between 256 – 65535.

Installing CGI Tools and SAS Foundation on Machines with Different Operating Systems

Your SAS Foundation system's operating system might be different than your CGI Tools system's operating system. For example, your SAS Foundation might be installed on a Windows system and your CGI Tools might be installed on a UNIX system. The CGI Tools install from the SDW will detect the destination operating system and install the appropriate operating system-specific software.

There are two methods to make the SAS Software Depot available to the installer on the destination CGI Tools system. The method you choose is dependent on the facilities available at your site. To access a SAS Software Depot on the destination CGI Tools system, do one of the following:

1. Launch the set-up from a SAS Software Depot residing on a remote system. You might need to use a cross-platform file access method, such as NFS or SAMBA, to connect the two systems.
2. Create media from an existing depot using the SAS Deployment Wizard and use that media on the host machine. This process is described more thoroughly in the *SAS Deployment Wizard and SAS Deployment Manager 9.3: User's Guide*, available from Install Center (http://support.sas.com/documentation/installcenter/en/ikdeploywizug/64204/PDF/default/user.pdf).

Note: *SAS/IntrNet operation requires TCP/IP connectivity between the SAS Foundation system and the CGI Tools system regardless of which operating systems these components are installed on.*

Test the Web Server

To determine if the Web server is running, launch the Web server's browser and enter `http://localhost`. This will return a Web page if the Web server is running.

If you do not receive a web page, you must debug or reinstall your Web server before continuing.

Test the Application Broker

To verify that CGI Tools was installed correctly and can access the `broker.cfg` file, point your Web browser to the following URL:

Windows:

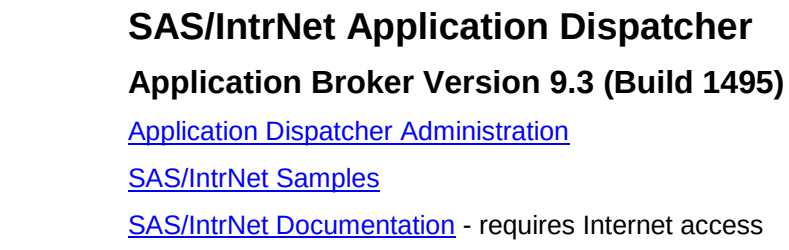
IIS - `http://your_webserver/scripts/broker.exe`

Apache - `http://your_webserver/cgi-bin/broker.exe`

Other hosts:

`http://your_webserver/cgi-bin/broker`

Replace `your_webserver` with the name of the Web server. The URL path might also need to be changed if you installed CGI Tools to a different directory. You should see a Web page similar to the following:



If you do not receive this page, you must debug your Web server installation before continuing. Verify that your Web server is enabled for CGI execution in the directory where you installed the Application Broker (broker.exe and broker.cfg files). This directory was determined by what was

entered for **Physical path for SAS/IntrNet CGI Executables** during the CGI Tools installation above.

Configure a Socket Service

To create the default service for an Application Server running in a UNIX environment, perform the following steps:

1. From a system prompt, submit the following command:

   ```
   SASHOME/SASDeploymentManager/9.3/sasdm.sh
   ```

   SASHOME is the path to the SAS home directory. The **Choose Language** window appears. Select the desired language and click **OK**.

2. The SAS Deployment Manager will display the **Select SAS Deployment Manager Task** window. Under **SAS/IntrNet Service Tasks** select **Create Socket Service** and click **Next** to continue.

3. The **Specify Service Name** window displays. The default value for the Service Name field is *default*. Create this as your first service because this is what is used when you run the samples. Click **Next** to continue.

4. The **Specify Service Directory** window displays. The SDM selects a default service root directory based on the location that you chose for user files when you installed SAS software. This default location is recommended for most users, although you can use the **Browse** button to select a different directory. Remember this directory because the start.sh script to start the Application Server will be created in it. Click **Next** to continue.

5. The **Specify Service Ports** window displays. Type the TCP/IP port number that you reserved for the default Application Dispatcher service. Click **Next** to continue.

6. The **Specify Administrator Password** window displays. A password is not necessary for the default service. You can add an administrator password later if you use this service for production applications. Click **Next** to continue.

7. The **Summary** window displays. It will say **Stage 1: Create Socket Service**. Use the **Back** button to go back and change any of the values you entered previously. Click **Start** when you are satisfied that the information you have entered is correct.

8. The **In Progress** window displays while SDM creates the service.

9. The **Deployment Complete** window displays when the task is finished. If the service is created correctly, a green checkmark will appear next to the **1. SAS/IntrNet** under **Stage 1: Create Socket Service**. If there was a problem a yellow exclamation or red X will appear and you should check the log for a description of the problem. The logs reside in *SASHOME*/SASFoundation/9.3/misc/intrnet.

10. The configuration utility created a start.sh file to start the default Application Server. Change to the service directory path and start the server by submitting the following command:

    ```
    ./start.sh
    ```

Starting the Socket Service

As stated above, change to the service directory path and start the server by submitting the following command: `./start.sh`

Testing the Socket Service

1. To make sure that the service was installed and started correctly, point your Web browser to this URL:

   Windows:

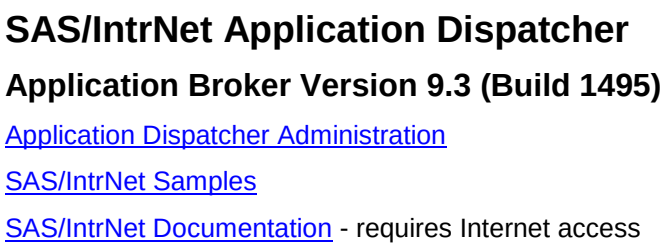
   IIS - `http://`*`your-webserver`*`/scripts/broker.exe`

   Apache - `http://`*`your-webserver`*`/cgi-bin/broker.exe`

   Other hosts:

   `http://`*`your-webserver`*`/cgi-bin/broker`

   Replace *`your-webserver`* with the name of the Web server. The URL path might also need to be changed if you installed the Application Broker to a different directory. You should see the following web page:

2. Click on the **Application Dispatcher Administration** link to see if the Application Broker can read the `broker.cfg` file. The Application Dispatcher Services Web page should open.
3. Verify connectivity between the Application Server and the Web server. Click on the **Application Dispatcher Administration** link and then click on the **ping** link under **SocketService default** heading. If the ping is successful, you should see:

   ```
   Ping. The Application Server host-name:port is functioning
   properly.
   ```

4. To complete installation testing, type this URL in your browser address line:

   Windows:

   IIS - `http://`*`your-webserver`*`/scripts/broker.exe?_service=default&_program=sample.webhello.sas`

   Apache - `http://`*`youriwebserver`*`/cgi-bin/broker.exe?_service=default&_program=sample.webhello.sas`

   Other hosts:

   `http://`*`your-webserver`*`/cgi-bin/broker?_service=default&_program=sample.webhello.sas`

   You should see the string “Hello World!” in large bold type in your browser. If you do not, add the debug option to create a log:

Windows:

IIS - `http://your-webserver/scripts/broker.exe?_service=default&_program=sample.webhello.sas&_debug=131`

Apache - `http://your-webserver/cgi-bin/broker.exe?_service=default&_program=sample.webhello.sas&_debug=131`

Other hosts:

`http://your-webserver/cgi-bin/broker?_service=default&_program=sample.webhello.sas&_debug=131`

Save the log screen on the browser for SAS Technical Support.

Configure Additional Services

This chapter only describes how to configure a simple default Application Dispatcher service. There are many reasons you may want to configure additional services, including segregating applications by security or performance requirements and implementing more scalable servers. See the "Using Services" section of the SAS/IntrNet Application Dispatcher documentation at http://support.sas.com/documentation/onlinedoc/intrnet/index.html for information on configuring additional services, using the Load Manager, and adding pool services.

Chapter 15 – Post-Installation Configuration for SAS/SECURE Software

SAS/SECURE software includes client components that non-SAS System client applications can use to communicate with a SAS server in a secure environment. To use encryption between a non-SAS System client and a SAS server with SAS/SECURE software licensed, you must install the SAS/SECURE client components on the client machine.

Note: *This installation is not necessary if the SAS System is your client. The SAS System installs the components that it needs as part of the SAS System install process.*

SAS/SECURE Client for Windows

The SAS/SECURE components needed by Windows clients can be installed by running the SAS Deployment Wizard.

SAS/SECURE Client for Java

The SAS/SECURE components for Java clients provide encryption support for Java applications. You can incorporate this support into applications that are written using the following components:

- SAS/SHARE driver for JDBC
- SAS/CONNECT driver for Java
- IOM Bridge for Java

The SAS/SECURE components needed by Java clients can be installed by running the SAS Deployment Wizard. The SECUREJAVA folder contains two JAR files that enable Java clients to use the CryptoAPI algorithms:

- `sas.rutil.jar` - should be copied to the location where the client you are running gets started.
- `sas.core.jar` - included in case you do not already have one however, this will most likely not be needed.

FIPS-Compliant Encryption

FIPS stands for Federal Information Processing Systems, and its 140-2 standard defines the security requirements for cryptographic modules. The 140-2 standard is detailed in the following document: `http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf`.

The ENCRYPTFIPS option is an option added to SAS/CONNECT spawners, IOM spawners, and Base SAS so that the communication encryption will be compliant with FIPS 140-2.. The ENCRYPTFIPS option is required for FIPS-compliant encryption.

The only requirement if you use ENCRYPTFIPS is that the NETENCRALG option must be set to AES or SSL. By default, the UNIX SSL module is not FIPS-compliant. If you want a FIPS-compliant OpenSSL module, you must download the OpenSSL source code that is FIPS-compliant, compile it, and use it to replace the OpenSSL libraries shipped with SAS/SECURE.

For more information about FIPS and encryption in general, refer to *Encryption in SAS 9.3*, available at
http://support.sas.com/documentation/onlinedoc/base/index.html#base93.

Chapter 16 – Post-Installation Configuration for SAS/SHARE Software

User Authentication

You are required to complete the steps from the chapter "Post-Installation Configuration for User Authentication and Identification" above. This allows SAS/SHARE software to authenticate a client's identity and check a client's authority to access resources.

System Configuration for the TCP/IP Communications Method

We suggest each SAS/SHARE server that runs on a network node be defined as a service in the file `/etc/services` or `/etc/inet/services` on that node. Each entry in this file associates a service name with the port number and protocol used by that service. An entry for a SAS/SHARE server has the following form:

server-name *port-number*/tcp # *comments*

The server name must be one to eight characters in length. The first character must be a letter or underscore; the remaining seven characters can include letters, digits, underscores, the dollar $ sign, or the at @ sign. The port number must be above 1024, as any port number equal to or less than 1024 is reserved.

An entry for a server whose name is `MKTSERV` might look like the following:

```
mktserv     5000/tcp    # SAS/SHARE server for Marketing and Sales
```

The server name is specified with the `SERVER=` option in the `LIBNAME` statement, in the `OPERATE`, and in the `SERVER` procedure. If a server name is not defined in the services file, you must specify "*__port*", two consecutive underscores followed by the port number (e.g., `server=__5012)`.

Client Components

SAS/SHARE software includes client components that are used outside of your SAS installation. These components are available from the SAS 9.3 Software Download site and are described below.

SAS/SHARE Data Provider

The SAS/SHARE Data Provider enables you to access, update, and manipulate SAS data using OLE DB- and ADO-compliant applications on Windows platforms.

SAS ODBC Driver

The SAS ODBC Driver enables you to access, update, and manipulate SAS data from ODBC-compliant applications on Windows platforms.

SAS/SHARE Driver for JDBC

The SAS/SHARE Driver for JDBC enables you to write applets, applications, and servlets that access and update SAS data. The Java Tools package that includes the SAS/SHARE driver for JDBC also includes the SAS/CONNECT driver for Java. If you are writing Java programs using these interfaces, you may also want to use the tunnel feature. This optional feature can be used with the Java applets you write to solve some common configuration problems.

SAS/SHARE SQL Library for C

The SAS/SHARE SQL Library for C provides an application programming interface (API) that enables your applications to send SQL queries and statements through a SAS/SHARE server to data on remote hosts.

NLS Information

Sites that develop or support international applications that use SAS/SHARE software should refer to Chapter 8, “Post-Installation Configuration for National Language Support (NLS).”

Chapter 17 – Using Host Sort Routines

This chapter provides instructions for making host sort routines available to SAS 9.3. The only supported host sort routine is SyncSort. To use host sort routines with SAS 9.3, complete the following steps:

1. Install the host sort library on your system by following the instructions provided by the vendor. Ensure that the host sort routine works outside of SAS 9.3.
2. Make the host sort library available to SAS 9.3 by following the instructions in the following section, "Making Host Sort Routines Available."
3. Submit an options statement in a SAS session to specify the host sort routine by following the instructions in the section "Using Host Sort Routines in a SAS Session."

Note: *For information on using host sort routines in a SAS session once they are available, refer to the* SAS 9.3 Companion for UNIX Environments.

Making Host Sort Routines Available

This section describes the system-specific instructions for making host sort routines available to SAS 9.3.

For AIX

Set the environment variable $LIBPATH to the directory containing the host sort library. For example, if the directory is `/usr/local/syncsort/lib`, then add these lines to both `!SASROOT/bin/sasenv_local` and `!SASROOT/bin/sasenv_local.ksh`:

```
LIBPATH=/usr/local/syncsort/lib:$LIBPATH
export LIBPATH
```

Add this line to `!SASROOT/bin/sasenv_local.csh`:

```
setenv LIBPATH /usr/local/syncsort/lib:$LIBPATH
```

For Linux and Solaris

Set the environment variable $LD_LIBRARY_PATH to the directory containing the host sort library. For example, if the directory is `/usr/local/syncsort/lib`, then add these lines to both `!SASROOT/bin/sasenv_local` and `!SASROOT/bin/sasenv_local.ksh`:

```
LD_LIBRARY_PATH=/usr/local/syncsort/lib:$LIBPATH
export LD_LIBRARY_PATH
```

Add this line to `!SASROOT/bin/sasenv_local.csh`:

```
setenv LD_LIBRARY_PATH /usr/local/syncsort/lib:$LIBPATH
```

For HP-UX

Set the environment variable $SHLIB_PATH to the directory containing the host sort library. For example, if the directory is `/usr/local/syncsort/lib`, then add these lines to both `!SASROOT/bin/sasenv_local` and `!SASROOT/bin/sasenv_local.ksh`:

```
SHLIB_PATH=/usr/local/syncsort/lib:$LIBPATH
export SHLIB_PATH
```

Add this line to `!SASROOT/bin/sasenv_local.csh`:

```
setenv SHLIB_PATH /usr/local/syncsort/lib:$LIBPATH
```

Using Host Sort Routines in a SAS Session

Note: *The options statements throughout this section specify the syntax to submit to the SAS System. You can also specify these options as command line options and options in the sasv8.cfg file. Refer to the* SAS Companion for UNIX Environments *for more information on setting options.*

Use the SORTNAME option to tell the SAS System which host sort routine should be used. Submit one of the following options statements in a SAS session:

- To use SyncSort (the default):

  ```
  OPTIONS SORTNAME=SYNCSORT;
  ```

- To use CoSORT:

  ```
  OPTIONS SORTNAME=COSORT;
  ```

Once the host sort routine is available, use the `SORTPGM=HOST` or `SORTPGM=BEST` options statements to tell the SAS System when to use the host sort routine.

Submit one of the following options statements in a SAS session:

- `OPTIONS SORTPGM=HOST;`

  tells the SAS System to always use the host sort routine made available.

- `OPTIONS SORTPGM=BEST;`

  tells the SAS System to choose the best sorting method in a given situation, the SAS System sort or the host sort.

  There are two options that define how the SAS System chooses the "best" sort algorithm. The following examples use the syntax of an options statement that needs to be submitted to the SAS System:

  - `-sortcut` *`n`*, where `n` specifies a number of observations.

    ```
    OPTIONS SORTPGM=BEST SORTCUT=500;
    ```

    `-sortcut` tells the SAS System to choose the host sort routine if the number of observations is greater than the number you specify, and to use the SAS System sort if the number of observations is equal to or less than the number specified.

  - `-sortcutp` *`size`*`[kKmM]`, where *`size`* specifies a file size in either kilobytes or megabytes.

    ```
    OPTIONS SORTPGM=BEST SORTCUTP=40M;
    ```

`-sortcutp` tells the SAS System to choose the host sort routine if the size of the data being sorted exceeds the size you specify, and to use the SAS System sort if the size of the data is equal to or smaller than the size you specify.

If these options are not defined or these options are set to zero, the SAS System chooses the SAS System sort routine. If you specify both options and either condition is met, the SAS System chooses the host sort routine.

You can change the work directory used for temporary sort files by using the option `sortdev` *`dir`*, where *`dir`* is the directory in which you want the temporary files to be created. For example, submit the following statement if you want the temporary files to be created in `/tmp`:

```
OPTIONS SORTPGM=BEST SORTCUT=500 sortdev="/tmp";
```

You can specify the host sort option `sortanom t` to print timing and resource information to the SAS log after each phase of a sort. The following is an example of this option:

```
OPTIONS SORTPGM=HOST SORTANOM=t;
```

You can specify the host sort option `sortanom v` to print to the SAS log the arguments passed to the sort, which may be useful for tuning or debugging:

```
OPTIONS SORTPGM=HOST SORTANOM=v;
```

You can attempt to increase your sort performance by increasing the values of the sortsize and memsize SAS options. However, make sure that sortsize is at least 4M less than memsize.

You can see other SAS performance statistics in the SAS log using the FULLSTIMER option:

```
OPTIONS FULLSTIMER;
```

support.sas.com

SAS is the world leader in providing software and services that enable customers to transform data from all areas of their business into intelligence. SAS solutions help organizations make better, more informed decisions and maximize customer, supplier, and organizational relationships. For more than 30 years, SAS has been giving customers around the world The Power to Know®. Visit us at **www.sas.com**.