CHAPTER

1

Concepts

Developing Windows Clients

When developing Microsoft Windows clients, you interact with the SAS Integrated Object Model (IOM) using the Microsoft Component Object Model (COM). In all the leading programming language products under Windows, and in most Windows applications, COM is the predominant mechanism for software interoperability on the Windows platform.

For the benefit of Windows applications, SAS manifests its IOM as a COM component that uses the automation type system. Microsoft calls this type of COM component an *ActiveX component* or an *OLE Automation server*.

Interacting with SAS as an ActiveX component has the following benefits:

- SAS can be called from a wide variety of programming language environments such as Microsoft Visual Basic (including Visual Basic for Applications and VBScript), Microsoft Visual C++, Borland C++Builder, Visual Basic .NET, and Visual C# .NET, Perl, Borland Delphi, and others.

- SAS processing can be invoked from the macro language of many popular applications, including those in Microsoft Office.
- The programming language skills most commonly used to build solutions in the Windows environment can also be applied to developing solutions that involve SAS.
- Operating system-level security, configuration and management are the same for SAS as for other applications and systems utilities.

In addition to these standard advantages for integration via COM, the SAS IOM offers a superior capability that is not commonly available to ActiveX components. This function, known as the IOM Bridge for COM, provides the ability to run the server on platforms other than Windows. Using this bridge, a Windows application can request SAS analytical processing for data on a UNIX or z/OS server and receive the results.

The exact interfacing technique used by Windows language products has evolved over the years. The initial approach that was documented in COM supported calls from Visual Basic by using an interface known as IDispatch. With IDispatch, calls into an interface go through a single method (IDispatch:Invoke), and then the appropriate implementation code is looked up at run time. This technique was compatible with early versions of Visual Basic, but was not optimal because of the amount of run-time interpretation that is involved in a method call. To improve performance, subsequent versions of Visual Basic and other languages can use *v-table binding* to call methods directly. Besides yielding better performance, v-table binding is also the most natural approach for COM calls from C++. The IOM implementation of ActiveX component interfaces uses the dual interface layout that provides both IDispatch and v-table binding. This dual interface gives the best performance from newer language implementations, but still supports the Dispatch technique for client languages (including VBScript and JScript) that use the older approach.

SAS®9 Integration Technologies includes a new client-side component called the *object manager*. The SAS 8 *workspace manager* is still supported, but it is recommended that you use the object manager interface in order to take advantage of the new features.

If you are using a SAS Metadata Server, then the object manager allows you to launch and manage objects on other SAS Metadata Servers, SAS Workspace Servers, SAS Stored Process Servers, and SAS OLAP Servers.

Client Requirements

The SAS Integration Technologies client for the Windows operating environment has the following software requirements:

- Windows XP, Windows Vista, Windows Server 2003, or Windows Server 2008. The client can connect to SAS using all three methods (COM, Distributed Component Object Model [DCOM], IOM Bridge for COM).
- If your client machines use the IOM Bridge for COM (a component of SAS Integration Technologies) to attach to an IOM server, then each client machine needs a valid TCP/IP configuration.
- If you are using the IOM Data Provider (a component of SAS Integration Technologies), then you need the Microsoft Data Access Components (MDAC) Version 2.1 and later on each client machine. MDAC is available from the Microsoft Web site.

Client Installation

Installation Methods

The SAS Integration Technologies client for the Windows operating environment can be installed in multiple ways:

- Install Base SAS for Windows. The SAS Integration Technologies client is installed with Base SAS software.
- Install an enterprise client (such as SAS Enterprise Guide). The setup program for the enterprise client component installs the SAS Integration Technologies client.
- Install the SAS Integration Technologies client by itself. The SAS Integration Technologies client is packaged into a single executable file that can be copied to Windows machines for easy installation of the SAS Integration Technologies client.

If you have not previously installed SAS 8.2 or earlier, the default installation directory is **`C:\Program Files\SAS\Shared Files`**.

If you have previously installed SAS 8.2 or earlier, the default installation directory is **`C:\Program Files\SAS Institute\Shared Files`**.

Note: If you are using the SAS Integration Technologies client with 64-bit SAS, then additional setup steps are required for IOM COM servers on 64-bit Windows. For details, see the SAS installation documentation. △

Components of the SAS Integration Technologies Client

Regardless of which method is used to install the SAS Integration Technologies client, the core function is installed via the inttech.exe file. When this file executes, it unbundles and installs the following components:

- the Integration Technologies Configuration Utility (itconfig.exe)
- SAS Package Reader (SASspk.dll, with Help in sasspk.chm)
- IOM Bridge for COM (SASComb.dll)
- Object Manager (SASOMan.dll, help file in sasoman.chm)
- Workspace Manager (SASWMan.dll, help file in saswman.chm)
- SAS Stored Process Service (sassps.dll, help file in sassps.chm)
- SAS Logging Service (LoggingService.dll)
- SAS Metadata Service (SAS.Metadata.dll)
- SAS IOM Data Provider
- SAS
- SAS type libraries (sas.tlb with Help in SAS.chm, arbor.tlb, asp.tlb, gms.tlb, IMLPlusServer.tlb, LoggingService.tlb, mdx.tlb, mqx.tlb, ObjectSpawner.tlb, olap.tlb, omi.tlb, sascache.tlb, SASIOMCommon.tlb, sastableserver.tlb, stp.tlb, tst.tlb)
- an executable to register type libraries (RegTypeLib.exe)
- Scripto (Scripto.dll)

- the Encryption Algorithm Manager (tcpdeam.dll) and the SAS Proprietary Encryption Algorithm (tcpdencr.dll), which are used by the IOM spawner and the IOM Bridge for COM components for encrypting the communication across the network
- the Xerces library from Apache Software Foundation

Note: The SAS Integration Technologies Windows client includes software that was developed by the Apache Software Foundation. For more details, see the Apache Web site at **www.apache.org**. △

Encryption Support

Windows clients (and Windows servers) that use strong encryption need additional support, which is supplied through SAS/SECURE software. SAS/SECURE software enables SAS Integration Technologies to use encryption algorithms that are available through the Microsoft Cryptographic Application Programming Interface (CryptoAPI).

If you have licensed SAS/SECURE software, you should install the SAS/SECURE client for Windows. You can install the SAS/SECURE client from the SAS Installation Kit.

The SAS/SECURE client installs the *tcpdcapi.dll* file that is necessary for the IOM Bridge for COM to use the CryptoAPI algorithms to communicate with the IOM server. The file is installed to the shared file area on the client.

For more information, see the online product overview for SAS/SECURE software at **www.sas.com/products/secure** on the SAS Products and Solutions Web site.

Windows Client Security

Security Contexts

Beginning in SAS®9, separate client and server security contexts within the SAS Workspace Server are not supported. All file access checks are now performed solely with the user ID under which the server was launched. The additional level of security checking, which was based on the client user ID, has been removed.

This change affects sites that were relying on two levels of access checking, one under the ServerUserID and an additional level under a ClientUserID that can be distinct. Before upgrading to SAS®9 from SAS 8, the following sites should review their security policies:

- sites that launch COM workspace servers by using the "This user" identity setting in the dcomcnfg utility
- sites that use COM+ pooling
- sites that use Web applications configurations where the site administration has control of the client security settings

IOM Bridge for COM Security

Specifying the ServerUserID

When using the IOM Bridge for COM, the spawner uses the client-provided login name and password to launch the server. The server receives its ServerUserID (OS process user ID) from the login name that is provided by the client. The operating system then performs access checks by using the ServerUserID.

Specifying Encryption

The IOM Bridge for COM supports encryption of network traffic between the client and the IOM server. Weak encryption support (through the SASPROPRIETARY encryption algorithm) is available with Base SAS software. Stronger encryption requires a license to SAS/SECURE on the server machine.

On the Windows platform, the SAS/SECURE license allows the encryption algorithms that are available through the CryptoAPI to be used. On other platforms, encryption algorithms are included in the SAS/SECURE software.

Note: The SAS/SECURE Client for Windows must be installed on the Windows client machines in order to use the CryptoAPI. (See "Client Installation" on page 3.) △

When you define a ServerDef, you can use the BridgeEncryptionAlgorithm and BridgeEncryptionLevel attributes to specify the encryption algorithm and the types of information that are encrypted. For more information, see the Object Manager package documentation (sasoman.chm.)

COM and DCOM Security

Specifying the ServerUserID

On Windows NT, the COM Service Control Manager (SCM) is responsible for launching COM and DCOM processes. The SCM reads values from the Windows registry to determine which identity to use when launching a process. These registry settings are configured by using the Windows dcomcnfg utility on the server.

The ServerUserID is set under the **`Identity`** tab for the properties of the *SAS.Workspace* application.

You can choose one of the following options for the identity to use to run the IOM server (the ServerUserID):

Interactive User allows the interactive user to kill the SAS server process through the task manager, because the interactive user owns the process. This is a security risk for the interactive user, because SAS will be running under this user's ID. Someone must be logged on, or the SAS process will not run.

Note: For SAS®9 and later, there is no longer an additional level of security checking that is based on the client user ID. All file access checks are performed based solely on the user ID of the interactive user. This setting is not recommended for production environments. △

Launching User specifies the default option and is more secure than the other two identity settings. This option ensures that the ServerUserID is the same as the client who created it. This setting provides a single-signon environment for launching the server; however, it does require the use of network authentication to set up the identity of the server. The Windows NT LAN Manager (NTLM) network authentication mechanism, which is the default if any Windows NT4 machines are involved or if Windows 2000 Kerberos is not set up, cannot pass the client identity across more than one machine boundary. If you configure IOM servers to use "launching user" for DCOM connections, then the servers cannot access network files unless special Windows registry settings are adjusted on the file server. The servers also cannot deliver events back to a DCOM client unless one of the following is true:

- The client permits everyone to access its COM interfaces by setting the authentication level to **`Everyone`**.
- The client turns off authentication and authorization by setting the authentication level to **`None`**.

This user allows you to configure a specific user name and password to run the IOM server. This option has the same security considerations as the interactive user, but you do not have to worry about always having someone logged on. Also, the identity does not change based on who is logged on.

Note: For SAS®9 and later, there is no longer an additional level of security checking based on the client user ID. All file access checks are performed based solely on the user ID that you specify for "This user." △

Specifying Encryption

For DCOM, encryption is enabled by using an authentication level of **`Packet Privacy`**.

Authentication and Impersonation Levels for COM and DCOM Security

Authentication levels must be set on both the client and server machines. The stronger of the two authentication levels is then selected. Here are the available authentication levels:

None The client is not authenticated. It is not possible for the server to determine the identity of the caller.

Connect The client is authenticated when the connection is first established and never again.

Call The client is authenticated each time a method call is made.

Packet Client authentication occurs for each packet. There might be multiple network packets used for a given call.

PacketIntegrity Each packet is authenticated and verified to have the same content as when it was sent.

PacketPrivacy All data is encrypted across the wire. Note that for local COM, the authentication level appears to be **`PacketPrivacy`**, but no encryption actually occurs on the local machine.

For more information about authentication levels, see the Microsoft documentation.

The impersonation level that is set on the client machine determines the impersonation level to use for the connection. The impersonation level that is set on the server machine is not used. Here are the available impersonation levels:

Anonymous	Impersonation is not allowed.
Identify	The server is allowed to know who is calling but cannot make calls by using the credentials of the caller.
Impersonate	The server can access resources by using the security credentials of the caller. The server cannot pass on the credentials. The IOM server attempts to impersonate the caller. This is the recommended setting.
Delegate	The server can access resources by using the security credentials of the caller and by passing on those credentials to other servers. SAS software does not currently support this option.

Programming Examples

This Visual Basic example retrieves the ClientUserID and the ServerUserID from the IOM server.

```
Public Sub sectest()
' This example prints both the client user ID and the server user ID.
' In SAS 9, the ServerUserID and the ClientUserID will always be the same.
' Create a local COM connection.
Dim sinfo As String
Dim swinfo() As String
Dim hwinfo() As String
Dim obWSMgr As New SASWorkspaceManager.WorkspaceManager
Set obSAS = obWSMgr.Workspaces.CreateWorkspaceByServer(
"MyServer", VisibilityNone, Nothing, "", "", sinfo)
' Get the host properties.
obSAS.Utilities.HostSystem.GetInfo swinfo, hwinfo
Debug.Print "ServerUserID: " & swinfo(
SAS.HostSystemSoftwareInfoIndexServerUserID)
Debug.Print "ClientUserID: " & swinfo(
SAS.HostSystemSoftwareInfoIndexClientUserID)
obSAS.Close
End Sub
```

COM Security Considerations for Client Applications

Here are some additional points to consider when developing client applications:

- Always test your application and configuration before making sensitive information available to ensure that people who are not authorized cannot see the data.
- Security settings and performance are inversely related. In general, the stronger the security, the slower things run. Security settings are highly configurable to allow the administrator to optimize performance for the required level of security.
- No system is completely secure, even at the strongest security settings.

- For maximum security when using the IOM Bridge for COM, use a BridgeEncryptionLevel of **`All`** and a strong BridgeEncryptionAlgorithm, such as RC4. For maximum performance, use a BridgeEncryptionLevel of **`None`**. (In this case, the setting for the BridgeEncryptionAlgorithm is ignored.)
- In general, for maximum security with DCOM, use an impersonation level of **`Impersonate`** and an authentication level of **`Packet Privacy`**.

  In SAS®9 and later, impersonation is not applied to workspace servers. For other IOM servers, an impersonation level of **`Impersonate`** is required.

  In SAS®9 and later, the use of connectionless transports (mainly UDP) can cause difficulty with configuring and debugging. If your system (particularly Windows NT4) still uses a connectionless transport, then you might avoid complications by naming a connection-oriented transport (typically TCP) as the primary default.

COM Security Settings for Threaded Multi-User IOM Servers

In SAS®9, SAS Integration Technologies provides three new threaded multi-user IOM servers. Windows Component Services has an entry for each application as follows:

SASMDX.Server
: SAS OLAP Server

SASOMI.OMI
: SAS Metadata Server

StoredProcessServer.StoredProcessServer
: SAS Stored Process Server, which provides interfaces to run user-written SAS programs (stored processes) to produce HTML output

If you do not specify the NOSECURITY object server parameter, then clients are authenticated when they connect to the server.

You can set DCOM security settings for each type of server individually. Use the Windows Component Services utility to specify security settings. In the Component Services utility, after you view the properties of a server, several tabs provide controls to customize the server's security.

Identity
: controls the ServerUserID for COM launches. If you launch the server via COM (rather than via the object spawner), then the first client that connects must be a Windows client. For multi-user servers, select **`This user`** and specify a user ID and password under which the server will run. The server runs in its own logon session; therefore, interactive logon and logoff activity on the same machine does not affect it. COM does not start the server with any environment variables and does not set a home directory. You should edit the SAS config file to remove environment variable references and to specify the "-sasinitialfolder" that you need at startup.

General
: controls the minimum client authentication level that the server accepts (which is also the level that the server uses on any outward calls). **`Connect`** (the default) is the minimum setting for the **`Authentication Level`**. Every COM client must also do the following:

  - define an authentication level of **`Connect`** or higher
  - set an impersonation level of **`Impersonate`**

These client settings can be specified by the client program on each calling interface or through one of two defaulting mechanisms:

- a call to CoInitializeSecurity() in the client program
- via the machine-wide default settings in COM security configuration

The ideal client program installs and uses an AppID of its own. However, some commonly used development languages, such as Visual Basic, do not provide an easy means to install and use an AppID.

Security
controls access, configuration, and launch permissions. These permissions can either be determined from machine-wide defaults or set up specifically for the particular IOM server application. Ensure that **`System`** is included in any of these permissions that you customize. You can use the access permissions editor to create a standard access control list to indicate who can use the server.

Location
indicates that the application should be run on this computer.

Endpoints
allows you to select the most used protocol. It is recommended that you use connection-oriented protocols such as TCP.

Windows Client Logging

Overview of Logging for Windows Clients

Logging for SAS Integration Technologies Windows clients is implemented by using the log4net framework from Apache. The log4net framework enables you to manage the logging for all of your Windows clients in a single logging configuration. The log4net configuration is stored in an XML file, which can be modified by using the Integration Technologies Configuration utility (ITConfig).

The log4net configuration consists of appenders and loggers. Appenders specify how the logging information is written to the log. Loggers associate a level in the object hierarchy with one or more appenders, and set the level of detail that is included in the log.

For more information about the log4net framework, see the Apache log4net site at **`http://logging.apache.org/log4net/`**.

Editing Your Logging Configuration

The logging configuration for Windows clients is stored in a file named log4netConfig.xml. If you configure logging for specific users, then the file is located the following path:

```
\Documents and Settings\user-name\Application Data\SAS\WindowsClientLogging
```

If you use a single configuration for all users, then the file is located in the following path:

```
\Documents and Settings\All Users\Application Data\SAS\WindowsClientLogging
```

To edit your configuration, perform the following steps:

1. Start the Integration Technologies Configuration utility by selecting **Start ▸ Programs ▸ SAS ▸ SAS 9.2 License Renewal & Utilities ▸ Integration Technologies Configuration**.
2. Select **`Configure Windows Client Logging`**, and then click **`Next`**.
3. Select whether you want to configure logging for the current user (**`Current user only`**), or for all users on this machine (**`All users of this machine`**). Click **`Next`**.
4. Select one of the following options:

   `Edit the current logging configuration file`
   specifies that you want to edit the current configuration.

   `Load default settings`
   specifies that you want to edit a new configuration based on the default settings.

   `Enable logging`
   if your configuration was previously disabled, then this option re-enables the current logging configuration and allows you to edit it.

   Click **`Next`**.
5. In the Configure Logging Appenders window, create the logging appenders that you want to use. For each appender, specify the following information:

   `Name`
   specifies the name of the appender.

   `Layout`
   specifies a formatting string that controls the output.

   `Type`
   specifies the type of appender. The appender type determines where the output is sent. For example, EventLogAppender sends output to the Windows Event Log. FileAppender writes output to a file.

   You can also click **`Filters`** to define filters for the appender. Filters enable you to send only the output that matches certain criteria. At the bottom of the list of filters, you must add a DenyAllFilter filter to remove the output that does not match the other filter.

   Depending on the appender type that you specify, additional information might be required. For more information about the information that you can specify, see the Integration Technologies Configuration Help.

   When you are finished defining appenders, click **`Next`**.
6. In the Configure Loggers window, create the loggers that you want to use. For each logger, the name specifies the logging namespace for the logger. You can select the standard SAS namespaces from the drop-down list or specify a custom namespace. The **`root`** logger specifies the default logging settings for Windows clients. For each logger, specify the level of logging detail and the appenders that are associated with the logger.

   If you want a logger to exclude its contents from any parent loggers, then deselect **`Enable Appender Additivity`**. For example, if you create a logger for SAS.BI and a logger for SAS.BI.Metadata, then deselect **`Enable Appender`**

`Additivity` for the SAS.BI.Metadata logger if you do not want its content to also appear in the SAS.BI logger.

For more information about specifying loggers, see the Integration Technologies Configuration Help.

When you are finished defining loggers, click **`Save`** to save your configuration file.

Disabling Windows Client Logging

To disable logging for SAS Integration Technologies Windows clients, perform the following steps:

1. Start the Integration Technologies Configuration utility by selecting **Start ▶ Programs ▶ SAS ▶ SAS 9.2 License Renewal & Utilities ▶ Integration Technologies Configuration**.
2. Select **`Configure Windows Client Logging`**, and then click **`Next`**.
3. Select whether you want to configure logging for the current user (**`Current user only`**), or for all users on this machine (**`All users of this machine`**). Click **`Next`**.
4. Select **`Disable logging`**, and then click **`Finish`**.