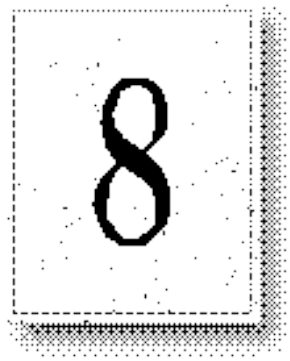
CHAPTER 1



Internet Information Server can publish both information and applications. This means your server can contain anything from static pages of information to interactive applications. You can also find and extract information from, and insert information into, databases.

This chapter explains how to:

- Prepare your server and your information for Publishing.
- Install and use interactive applications on your server.
- Publish by using an Open Database Connectivity (ODBC)–compliant data source.

Preparing Information for Publishing

Most Web pages are formatted in HyperText Markup Language (HTML). HTML files are simple ASCII text files with codes embedded to indicate formatting and hypertext links. HTML specifications are changing constantly. You should probably review the HTML specifications (available on the Internet) to fully plan your HTML pages.

Authoring HTML Files

You can use any text editor, such as Notepad or Write, to create and edit your HTML files; but you will probably find an HTML editor, such as Internet Assistant for Microsoft Word, easier to use.

You use the HTML editor or other system to create HTML files, which can include hyperlinks to other files on your system. If you want to include images or sounds, you will also need appropriate software to create and edit those files.

Publishing HTML and Other File Formats

Your files can include images and sound. You can even create links to Microsoft® Office files or to almost any other file format. Remote users must have the correct viewing application to view non-HTML files. For example, if you know that all remote users will have Microsoft Word, you can include links to Microsoft Word .doc files. The user can click the link and the document will appear in Word on the user's computer.

Once you have created your information in HTML or other formats, you can either copy the information to the default directory \Inetsrv\Wwwroot, or you can change the default home directory to the directory containing your information.

MIME Type Configuration

If your server provides files that are in multiple formats, your server must have a Multipurpose Internet Mail Extension (MIME) mapping for each file type. If MIME mapping on the server is not set up for a specific file type, browsers may not be able to retrieve the file. See the Windows NT Registry for the default MIME mappings.

To configure additional MIME mappings start Regedt32.exe and open

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters\MimeMap

Add the value for the MIME mapping required for your server with the following syntax:

```
<mime type>,<filename extension>,<unused parameter>,<gopher type>
```

For example:

```
text/html,htm,/unused,1
image/jpeg,jpeg,/unused,5
```

The default entry with the filename extension specified as an asterisk (*) is the default MIME type used when a MIME mapping does not exist. For example, to handle a request for the file Current.vgr when the the filename extension .vgr is not mapped to a MIME type, the server will use the MIME type specified for the asterisk extension, which is the type used for binary data.Usually, this will cause browsers to save the file to disk.

Including Other Files with the Include Statement

You can add repetitive information into an HTML file just before sending the file to a user. This feature is handy for including the same text on each HTML page, such as copyright information or a link to the home page.

The format of the include statement is:

```
<!--#include file="value"-->
```

The value must contain the full path, from the home directory of your WWW service.

For example, to include a link to your home page in each HTML document:

1. Create the file linkhome.htm, which contains the HTML codes you want to repeat; for example, a button to your home page. The file would contain HTML that looks similar to this:

   ```
   <A HREF="/homepage.htm"><IMG SRC="/images/button_h.gif"></A>
   ```

2. Use the filename extension .stm when you create your Web pages (rather than .htm or .html).
3. In each .stm file, use an **include file** statement where you want the repeated information to appear. For example:

   ```
   You can return to: <!--#include file="/linkhome.htm"--> at any time
   ```

Note that all paths are relative to the WWW home directory and can include virtual roots.

Publishing Dynamic Applications

One of the most exciting features of Microsoft Internet Information Server is the ability to run applications or scripts that remote users start by clicking HTML links or by filling in and sending an HTML form. Using programming languages such as C or Perl, you can create applications or scripts that communicate with the user in dynamic HTML pages.

Creating the Applications or Scripts

Interactive applications or scripts can be written in almost any 32-bit programming language, such as C or Perl, or as Windows NT batch files (.bat or .cmd). When you write your applications or script you can use one of two supported interfaces, the Microsoft Internet Server Application Programming Interface (ISAPI) or the Common Gateway Interface (CGI). Documentation for ISAPI is available from Microsoft via subscription to the Microsoft Developer Network (MSDN). Documentation for CGI is available on the Internet. Batch files can issue any command valid at the command prompt.

Applications that use ISAPI are compiled as Dynamic-Link Libraries (DLLs) that are loaded by the WWW service at startup. Because the programs are resident in memory, ISAPI programs are significantly faster than applications written to the CGI specification.

Internet Server API

ISAPI for Windows NT can be used to write applications that Web users can activate by filling out an HTML form or clicking a link in an HTML page on your Web server. The remote application can then take the user-supplied information and do almost anything with it that can be programmed, and then return the results in an HTML page or post the information in a database.

ISAPI can be used to create applications that run as DLLs on your Web server. If you have used Common Gateway Interface (CGI) scripts before, you will find that the ISAPI applications have much better performance because your applications are loaded into memory at server run-time. They require less overhead because each request does not start a separate process.

Another feature of ISAPI allows pre-processing of requests and post-processing of responses, permitting site-specific handling of HyperText Transport Protocol (HTTP) requests and responses. ISAPI filters can be used for applications such as customized authentication, access, or logging.

You can create very complex sites by using both ISAPI filters and applications. ISAPI extensions can also be combined with the Internet Database Connector to create highly interactive sites.

For complete information about programming with ISAPI, see the Microsoft BackOffice Software Development Kit (SDK), available from Microsoft. See the introductory chapter, “Before You Begin,” for further information about obtaining the ISAPI SDK.

Common Gateway Interface

The Common Gateway Interface (CGI) is a standard interface used to write applications that remote users can start by filling out an HTML form or clicking a link in an HTML page on your Web server. As with ISAPI, the remote application can then take the user-supplied information and do almost anything that can be programmed, then return the results of the application in an HTML page or post the information in a database. Because simple CGI applications are often written using scripting languages such as Perl, CGI applications are sometimes referred to as “scripts.”

Most 32-bit applications that run on Windows NT and conform to the CGI specifications can be used by Microsoft Internet Information Server.

For information about how to convert existing CGI programs or scripts from UNIX®, see Help.

For more information about the CGI specifications, consult the CGI specifications widely available on the Internet.

Installing Your Application on Internet Information Server

Once you have written your application or script, place it in the /Scripts directory, a virtual directory for applications. This virtual directory has Execute access.

You must also ensure that every process started by your application is running by using an account with adequate permissions. If your application interacts with other files, the account you assign to your program must have the right permissions to use those files. By default, applications run using the IUSR_*computername* account.

Running Your Application

If your application does not require data from the user, you will typically create a link to your application in a simple HTML file. If your application does require data from the user, you will probably use an HTML form. In other instances you can just send a Uniform Resource Locator (URL), usually containing data parameters, to invoke a program.

An HTML link to a application that does not require input from the user might look like the following example:

```
http://www.company.com/scripts/catalog.exe?
```

where \Scripts is the virtual directory for interactive applications.

If you are creating a application that requires input from the user, you will need to understand both HTML forms and how to use the forms with ISAPI or CGI. This information is widely available on the Internet or from other sources.

Associating Interpreters with Applications

Because you have the flexibility to create applications in almost any programming language, Internet Information Server uses the filename extension to determine which interpreter to invoke for each application. The default interpreter associations are listed below. You can use the Registry Editor to create additional associations as described in Help.

Extension	Default Interpreter
.exe, .com, .bat, .cmd	Cmd.exe
.idc	Httpodbc.dll

Security Implications

When you allow remote users to run applications on your computer, you run the risk of hackers attempting to break into your system. Microsoft Internet Information Server is configured by default to reduce the risk of malicious intrusion by applications in two important ways.

First, the virtual directory \Scripts contains your applications and is marked as a application directory. Only an administrator can add programs to a directory marked as an application directory. Thus, unauthorized users cannot copy a malicious application and then run it on your computer without first gaining administrator access.

Second, if you have configured the WWW service to allow only anonymous logons, all requests from remote users will use the IUSR_*computername* account. By default, the IUSR_*computername* account is unable to delete or change files by using the Windows NT File System (NTFS) unless specifically granted access by an administrator. Thus, even if a malicious program were copied to your computer, it would be unable to cause much damage to your content because it will only have IUSR_*computername* access to your computer and files.

Publishing Information and Using a Database

With the WWW service and the Open Data Base Connectivity (ODBC) drivers provided with Internet Information Server, you can:

- Create Web pages with information contained in a database.
- Insert, update, and delete information in the database based on user input from a Web page.
- Perform other Structured Query Language (SQL) commands.

How the Internet Database Connector Works

Conceptually, database access is performed by Internet Information Server as shown in the following diagram.

Web browsers (such as Internet Explorer, or browsers from other companies such as Netscape) submit requests to the Internet server by using HTTP. The Internet server responds with a document formatted in HTML. Access to databases is accomplished through a component of Internet Information Server called the Internet Database Connector. The Internet Database Connector, Httpodbc.dll, is an ISAPI DLL that uses ODBC to gain access to databases.

The following illustration shows the components for connecting to databases from Internet Information Server.

Httpodbc.dll uses two types of files to control how the database is accessed and how the output Web page is constructed. These files are Internet Database Connector (.idc) files and HTML extension (.htx) files.

The Internet Database Connector files contain the necessary information to connect to the appropriate ODBC data source and execute the SQL statement. An Internet Database Connector file also contains the name and location of the HTML extension file.

The HTML extension file is the template for the actual HTML document that will be returned to the Web browser after the database information has been merged into it by Httpodbc.dll.

Installing ODBC and Creating System Data Sources

When the ODBC option is selected during setup, ODBC version 2.5 components are installed. This version of ODBC supports System DSNs (Data Source Names) and is required for using ODBC with Microsoft Internet Information Server.

System DSNs were introduced in ODBC version 2.5 to allow Windows NT services to use ODBC.

» **To install the ODBC drivers**

1. If you did not install the ODBC Drivers and Administration option, run Setup again by double-clicking the Internet Information Server Setup icon in the Microsoft Internet Server program group of Program Manager. You will need the Internet Information Server compact disc, or a network installation directory containing the complete contents of the compact disc.
2. Choose the OK button.
3. Choose the Add/Remove button.
4. Choose the OK button.
5. Select the ODBC Drivers and Administration option.
6. Choose the OK button.
7. The Install Drivers dialog box appears.

8. To install the SQL Server driver, select the SQL Server driver from the Available ODBC Drivers list box, and choose the OK button.

   Setup completes copying files.

Important

» **To create the system data sources**

1. Double-click the Control Panel icon in the Main program group of Program Manager.
2. Double-click the ODBC icon.

   The ODBC Data Sources dialog box appears.

   You may see other data sources in the list if you previously installed other ODBC drivers.

3. Choose the System DSN button.

Be sure to click the System DSN button. The Internet Database Connector will work only with System DSNs.

The System Data Sources dialog box appears.

4. Choose the Add button.

   The Add Data Source dialog box appears.

5. Select SQL Server from the list box and click OK. The ODBC SQL Server Setup dialog box will appear.

6. Enter the name of the data source.

   The data source name is a logical name used by ODBC to refer to the SQL Server driver and the actual server name on which SQL Server is running. You can also use the server name “(local)” if SQL Server is running on this computer. The data source name is used in Internet Database Connector files to tell Internet Information Server where to access the data.

   The server name, network address, and network library are specific to your installation. If you do not know what to enter in these controls, accept the defaults. To find out the details, click the Help button and find the section that describes your network.

7. Choose the OK button.

   The System Data Sources dialog box will be displayed again, but now will have the name of the data source displayed.

8. Choose the Close button to close the System Data Sources dialog box.
9. Choose the Close button to close the Data Sources dialog box.
10. Choose the OK button to complete the ODBC and DSN setup.

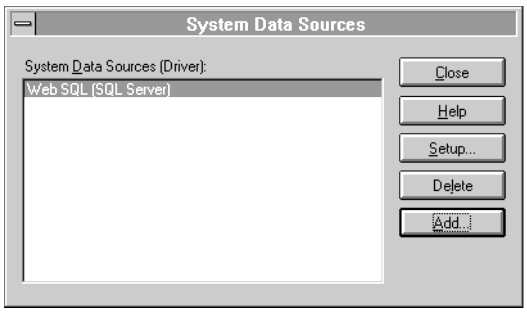
Authoring Web Pages with Database Access

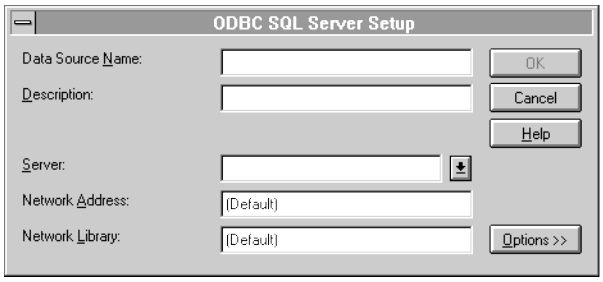
In order to provide access to a SQL database from your Web page, you will need to create an Internet Database Connector file and an HTML extension file.

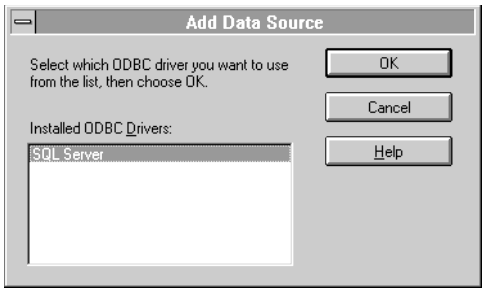
For more information on creating these files, see Help.

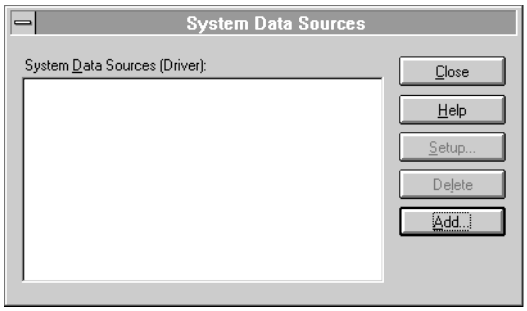
BLANK PAGE

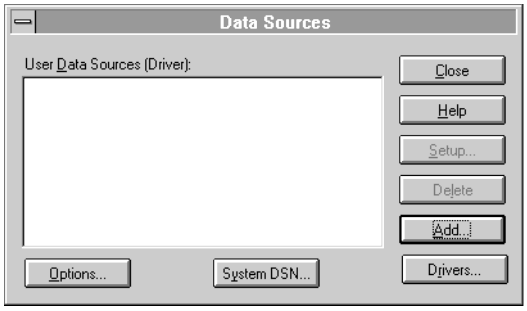
This page is intentionally left blank to preserve page numbering for the Table of Contents and Index. This text will appear on screen, but will not print on a PostScript printer.



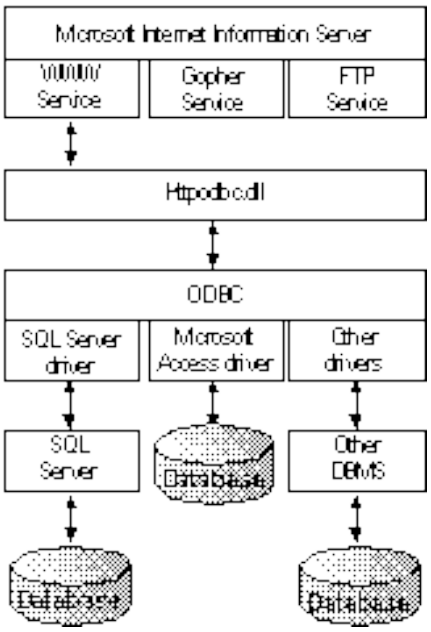


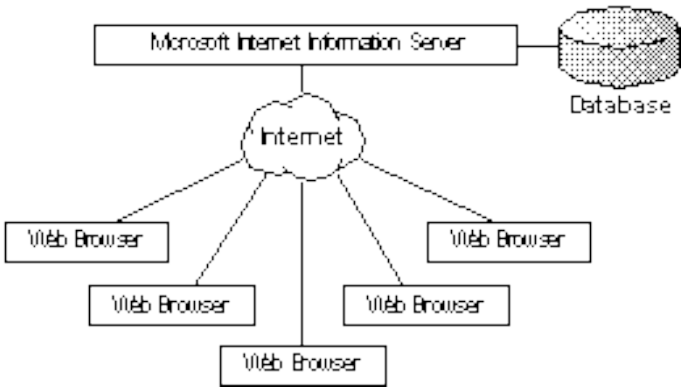


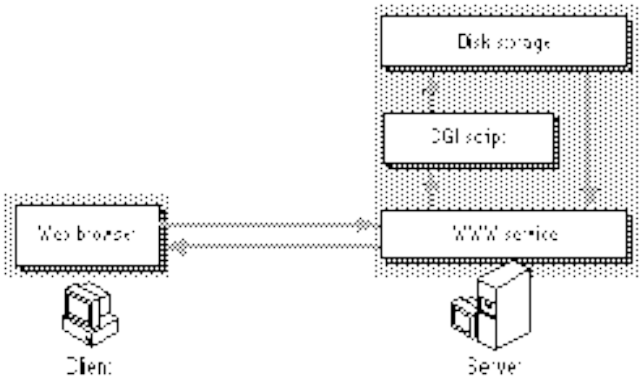

System Data Sources
System Data Sources (Driver):
Close
Help
Setup...
Delete
Add...

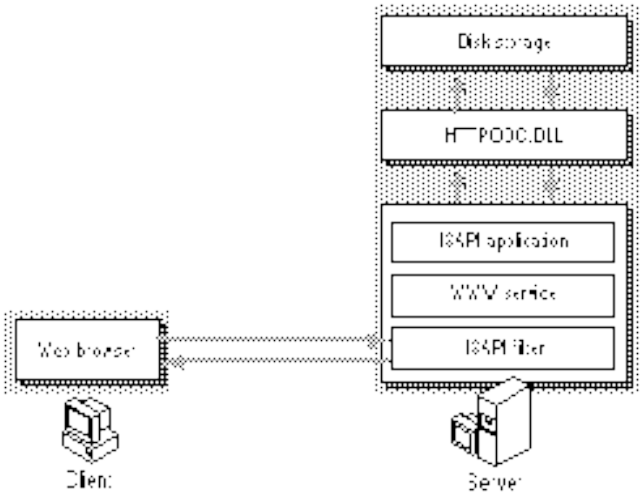

Data Sources
User Data Sources (Driver):
Close
Help
Setup...
Delete
Add...
Options...
System DSN...
Drivers...

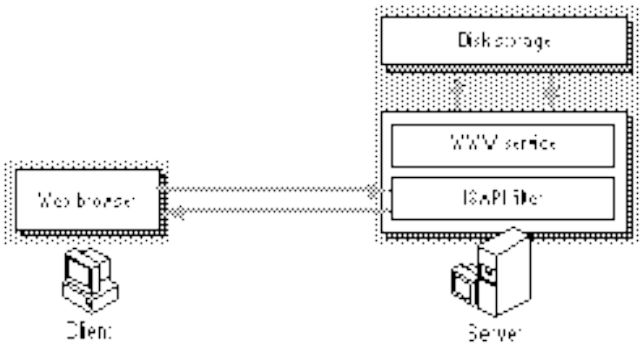
Install Drivers
Select one or more ODBC drivers to install from the list, then choose OK.
OK
Cancel
Help
Available ODBC Drivers:
SQL Server
Advanced...

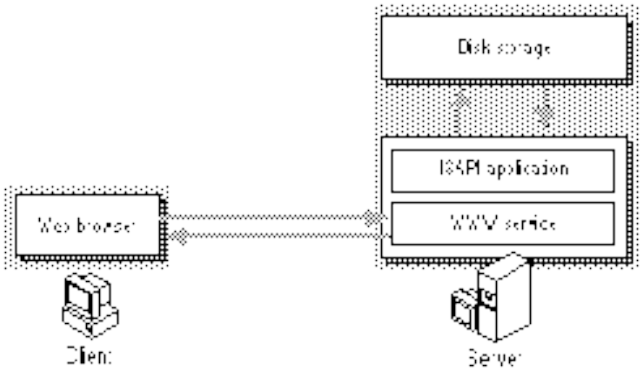

Microsoft Internet Information Server
WWW Service
Gopher Service
FTP Service
Httpodbc.dll
ODBC
SQL Server driver
Microsoft Access driver
Other drivers
SQL Server
Other DBMS


Microsoft Internet Information Server
Database
Internet
Web Browser
Web Browser
Web Browser
Web Browser
Web Browser


Disk storage
CGI script
Web browser
WWW service
Client
Server


Disk storage
HTTPODC.DLL
ISAPI application
WWW service
ISAPI filter
Web browser
Client
Server


Disk storage
WWW service
ISAPI filter
Web browser
Client
Server


Disk storage
ISAPI application
WWW service
Web browser
Client
Server