**WWW Service Properties**

You use the Service property sheet to control who can use your server and to specify the account used for anonymous client requests to log on to the computer. Most Internet sites allow anonymous logons. If you allow anonymous logons, then all user permissions for the user, such as permission to access information, will use the IUSR_*computername* account. To use your current security system to control information access, change the anonymous logon account from IUSR_*computername* to an existing account on your network.

This property sheet also sets the comment in the main Internet Service Manager window.

**Connection Timeout**

Sets the length of time before the server disconnects an inactive user. This value ensures that all connections are closed if the HTTP protocol fails to close a connection.

**Anonymous Logon**

Sets the Windows NT user account to use for permissions of all anonymous connections. By default, Internet Information Server creates and uses the account IUSR_*computername*. Note that the password is used only within Windows NT; anonymous users do not log on by using a username and password.

**Password Authentication**

Specifies the authentication process to use if anonymous access is disallowed or the remote client requests authentication.

Basic authentication is encoded. Basic authentication is typically used in conjunction with SSL to ensure that usernames and paswords are encrypted before tranmission. All browsers support Basic authentication.

Windows NT Challenge/Response automatically encrypts usernames and passwords. Internet Explorer version 2.0 and later supports this password authentication scheme.

**Comment**

Specifies the comment displayed in Internet Service Manager's report view.

**WWW Directories**

The WWW Directories property sheet sets directories and directory behavior for the WWW service.

**Directory listing box**

Lists the directories used by the WWW service.

**Directory** lists the path of directories used by the WWW service.

**Alias** is the path for service users for virtual directories.

**Address** lists the IP address using that directory.

**Error** indicates system errors, such as difficulty reading a directory.

**Add, Remove, and Edit buttons**

To set up a directory, press the Add button or select a directory in the Directories listing box and press the Edit button. The Remove button removes the directories that you select.

**Enable Default Document and Directory Browsing Allowed**

The Default Document and Directory Browsing settings in the Directories property sheet for the WWW service are used to set up default displays that will appear if a remote user does not specify a particular file. Directory browsing means that the user is presented with a hypertext listing of the directories and files so that the user can navigate through your directory structure. This function is similar to File Manager.

You can place a default document in each directory so that when a remote user does not specify a particular file, the default document in that directory is displayed. A hypertext directory listing is sent to the user if directory browsing is enabled and no default document is in the specified directory.

Note that virtual directories will not appear in directory listings; users must know a virtual directory's alias and type in its URL address, or click a link in an HTML page, to access virtual directories.

**WWW Directory Properties**

Configure the WWW service directories by using this dialog box. Press the Add button on the Directories property sheet to set up new directories.

**Directory**

Sets the path to the directory to use for the WWW service.

**Browse button**

Use to select the directory to use for the WWW service.

**Home Directory**

Choose this to specify the root directory for the WWW service.

Internet Information Server provides a default home directory, \Wwwroot, for the WWW service. The files that you place in the WWW home directory, and its subdirectories, are available to remote browsers. You can change the location of the default home directory.

**Virtual Directory**

Choose this to specify a subdirectory for the WWW service. Enter the directory name or "alias" that service users will use.

You can add other directories outside the home directory that are accessed by browsers as subdirectories of the home directory. That is, you can publish from other directories and have those directories accessible from within the home directory. Such directories are called "virtual directories."

The administrator can specify the physical location of the virtual directory and the virtual name (alias), which is the directory name used by remote browsers.

Note that virtual directories will not appear in WWW directory listings; you must create explict links in HTML files in order for users to access virtual directories. Users can also type in the URL if they know the alias for the virtual directory.

The published directories can be located on local or network drives. If the virtual directory is a network drive, provide the username and password with access to that network drive. Virtual directories on network drives must be on computers in the same Windows NT domain as the Internet Information Server.

**Account Information**

This box is active only if the Directory specified in the first line of this dialog box is a Universal Naming Convention (UNC) server and share name, for example, \\Webserver\Htmlfiles. Enter the username and password that has permission to use the network directory. Virtual directories on network drives must be on computers in the same Windows NT domain as the Internet Information Server.

**Important** If you specify a username and password to connect to a network drive, all Internet Information Server access to that directory will use that username and password. You should use care when using UNC connections to network drives to prevent possible security breaches.

**Virtual Servers (World Wide Web only)**

Select the Virtual Server check box and enter an IP address to create a directory for the virtual server. The IP address must be bound to the network card providing the service. Use the Network applet in Control Panel to bind additional IP addresses to your network card.

You can have multiple domain names on a single Internet Information Server–based computer so that it will appear that there are additional servers, or "virtual servers." This feature makes it possible to service WWW requests for two domain names (such as http://www.company1.com/ and http://www.company2.com/) from the same computer. Enter the IP (Internet Protocol) address for the home directory, and virtual directories for each virtual server that you will create.

If the path for a virtual directory is a network drive, provide a username and password with access to that network drive. Virtual directories on network drives must be on computers in the same Windows NT domain as the Internet Information Server.

**Important** If you have assigned more than one IP address to your server, when you create a directory you must specify which IP address has access to that directory. If no IP address is specified, that directory will be visible to all virtual servers. The default directories created during setup do not specify an IP address. You may need to specify IP addresses for the default directories when you add virtual servers.

**Access check boxes**

The Access check boxes control the attributes of the directory. If the files are on an NTFS drive, NTFS settings for the directory must match these settings.

**Read** must be selected for information directories. Do not select this box for directories containing programs.

**Execute** allows clients to run any programs in this directory. This box is selected by default for the directory created for programs. Put all your scripts and executable files into this directory. Do not select this box for directories containing static content.

**Require secure SSL channel** select this box if using SSL security to encrypt data transmissions.

**WWW Logging Properties**

The Logging property sheet sets logging  for the selected information service.

Logging provides valuable information about how a server is used. You can send log data to files or to an Open Data Base Connectivity (ODBC)–supported database. If you have multiple servers or services on a network, you can log all their activity to a single file or database on any network computer.

If you want to log to a file, you can specify how often to create new logs and which directory put the log files in. The Convlog.exe command prompt command converts log files to either EMWAC log files or the common log file format.

If you log to an ODBC data source, you must specify the ODBC Data Source Name (DSN), table, and valid username and password to the database.

**Enable Logging**

Select this box to start or stop logging for the selected information service.

**Log to File**

Choose this option to log to a text file for the selected information service.

**Automatically open new log**

Select this box to generate new logs at the specifed interval. If not selected, the same log file will grow indefinitely.

**Log file directory**

Shows the path to the directory containing all log files. To change directories, click Browse and select a different directory.

**Log file filename**

Names the log file. Lowercase letters **yy** will be replaced with the year, **mm** will be replaced with the month, and **dd** will be replaced with the day.

**Log to SQL/ODBC Database**

Choose to log to any ODBC data source. Set the Datasource name, Table name (not the filename of the table), and specify a username and password that is valid for the computer on which the database resides. You must also use the ODBC applet in Control Panel to create a system data source.

**WWW Advanced Properties**

The Advanced property sheet sets access by specific IP address to block individuals or groups from gaining access to your server. You can also set the maximum network bandwidth for outbound traffic, to control (throttle) the maximum amount of traffic on your server.

**IP Access Control**

You can control access to each Microsoft Internet Information Server service by specifying the IP address of the computers to be granted or denied access.

If you choose to grant access to all users by default, you can then specify the computers to be denied access. For example, if you have a form on your WWW server and a particular user on the Internet is entering multiple forms with ficticious information, you can prevent the computer at that IP address from connecting to your site. Conversely, if you choose to deny access to all users by default, you can then specify which computers are allowed access.

**Granted Access**

Choose this option, then press the Add button to list computers that will be denied access.

**Denied Access**

Choose this option, then press the Add button to list computers that will be granted access.

**Add**

To add computers that you want to deny access to, select the Granted Access button and click Add. Conversely, to add computers that you want to grant access to, select the Denied Access button, and click Add.

**Limit Network Use by all Internet Services on this computer**

You can control your Internet services by limiting the network bandwidth allowed for all of the Internet services on the server. Set the maximum kilobyes of outbound traffic permitted on this computer.

**WWW Grant or Deny Access**

Choose Single Computer and provide the IP Address to exclude a single computer. Choose Group of Computers and provide an IP Address and subnet mask to exclude a group of computers. Press the button next to the IP address to use a domain name system (DNS) name instead of IP address. Your server must have a DNS server specified in its TCP/IP settings.

You are specifying, by IP address or domain name, which computer or group of computers will be granted or denied access. If you choose to, by default, grant access to all users, you will specify the computers to be denied access. If you choose to, by default, deny access to all users, you will then specify the specific computers to be allowed access. You should fully understand TCP/IP networking, IP addressing, and the use of subnet masks to use this option.

**Internet Service Manager Authentication Options**

In addition to the "anonymous logon" username and password fields, the Service property sheet of Internet Service Manager contains the following authentication options:

**WWW**

- **Allow Anonymous** When this check box is selected, anonymous connections are processed, and the "anonymous logon" username and password are used for these connections. When this checkbox is unchecked, all anonymous connections are rejected, and basic or Windows NT Challenge/Response authentication protocol is required to access content.
- **Basic** When this check box is selected, the WWW service will process requests using basic authentication. **WARNING:** Basic authentication sends Windows NT usernames and passwords across the network without encryption. This check box is cleared by default for security reasons.
- **Windows NT Challenge/Response** When this check box is selected, the service will honor requests by clients to send user account information using the Windows NT Challenge/Response authentication protocol. This protocol uses a one-way hash to prevent passwords from being transmitted across the network. The Windows NT Challenge/Response authentication process is initiated automatically as a result of an "access denied" error on an anonymous client request. Currently, Windows NT Challenge/Response authentication protocol works only with Internet Explorer 2.0 for Windows 95.

**Note:** If the "Basic" and "Windows NT Challenge/Response" check boxes are both cleared (and the "Allow Anonymous" check box is selected), all client requests are processed as anonymous requests. In this case, if the client supplies a username and password in the request, this username and password are ignored by the WWW service. The "anonymous logon" user account will be used to process the request.

Help not available.

Help not available.

### Controlling Security by Windows NT Username and Password

#### To set username and password security

1 In Internet Service Manager double-click the service to display its property sheets, then click the Service tab.

2 In the Anonymous Logon box, type the username and password to be used by the WWW service when accessing resources on behalf of an anonymous client.

This account must be a valid account in the Windows NT User Manager.

3 In the Password Authentication box, choose the following options:

Allow Anonymous.

Basic (clear text)

Windows NT Challenge/Response for encrypted authentication. This option works only for clients using Internet Explorer version 2.0 (or later).

4 Click OK.

**Setting WWW Directory Access**

When creating a Web publishing directory in Internet Service Manager, three access permissions can be selected via checkboxes on the Directories dialog box. The setting of these access permissions applies to the defined home directory or virtual directory, and all of its subdirectories. The permissions are:

**Read**

In order for files stored in a directory (home directory or virtual directory) to be downloaded to a client, the directory's Read permission must be enabled. If a client sends a request for a file that is in a directory without Read permission, the Web service will return an error. Directories containing information to publish (HTML files, for example) would typically have this permission enabled. Directories containing CGI applications and Internet Server Application Program Interface (ISAPI) DLLs would typically have this permission disabled, to prevent clients from downloading the application files.

**Execute**

A client request can invoke a CGI application or an Internet Server Application Program Interface (ISAPI) application in one ofsd two ways:

The filename of the CGI executable or the ISAPI DLL can be specified in the request (URL). An example URL would be:

http://inetsrvr.microsoft.com/scripts/httpodbc.dll/scripts/pubs.idc?lname=Smith

For this request to be valid, the file Httpodbc.dll must be stored somewhere in the Web publishing tree (in this example, \Scripts), and the directory it is stored in must have the Execute permission selected. This way the administrator can permit applications (CGI or ISAPI) to be run from a small number of carefully monitored directories.

The other way to configure CGI and ISAPI applications is to use the Web File Extension Mapping feature, which allows your executables and DLLs to be stored somewhere other than the Web publishing tree. An example URL would be:

http://inetsrvr.microsoft.com/scripts/pubs.idc?lname=Smith

In this example, the script file is stored in a directory of the Web publishing tree that has the Execute permission enabled. The service, upon receiving the request, will use the filename-extension mappings to determine where to find the application, which can be stored anywhere. This technique prevents users from invoking CGI and ISAPI applications directly by adding parameters in the URL. This is therefore a more secure mechanism, and useful for all Web applications and scripts. Please see filename-extension mapping for more information.

**Require secure SSL channel**

This option must be selected to require encrypted communication for a directory. For more information on Secure Sockets Layer (SSL), see SSL.

**Setting Up a Virtual Server**

1 In Internet Service Manager, double-click the WWW Service.
2 Click the Directories tab.
3 Click the Add button.
4 In the Directory box of the Directory Properties dialog box, select a directory by clicking the Browse button.
5 In the Alias box, type the name you have chosen for your virtual server.
6 Select the Virtual Server check box.
7 Type the IP address for the virtual server.
This address is usually supplied by your Internet service provider.
8 Select the Execute option if you want to allow users to execute applications on your site.
9 Click OK.

### Setting Up CGI Applications

**To set up a Common Gateway Interface (CGI) application on your Web server**

1 Create the application, conforming to the CGI specification.

2 Put the application in the default /Scripts directory, or ensure Execute permission has been granted to the directory containing the CGI application in both Internet Service Manager and File Manager (on NTFS drives).

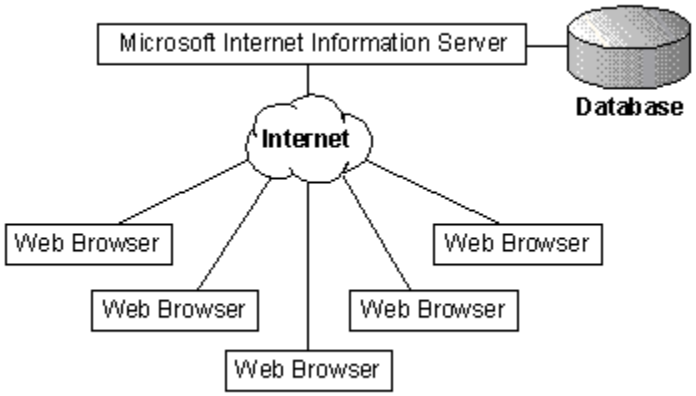
For better performance, Microsoft offers the Internet Server API for creating applications. For more information about the Internet Server API, see the print documentation.

**Internet Database Connector Overview**

With Internet Information Server and Open Data Base Connectivity (ODBC), you can:

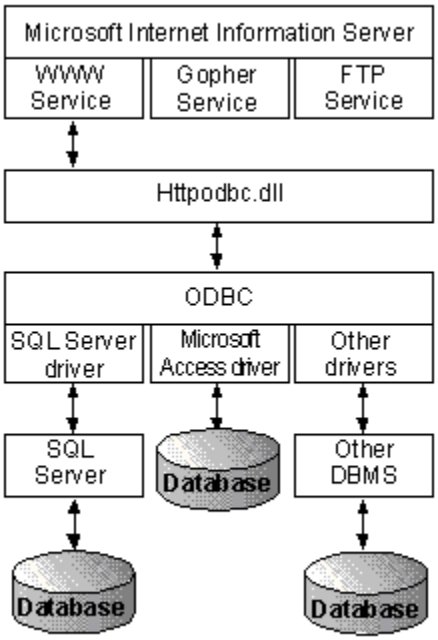
- Create Web pages with information contained in a database.
- Insert, update, and delete information in the database based on user input from a Web page.
- Perform other SQL commands.

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




Web browsers submit requests to the Internet server by using the HTTP protocol. The Internet server responds with a document formatted in HTML. This document is often referred to as a "Web page. "Retrieving data from 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 retrieve data from 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 desired ODBC data source and execute the SQL statement. A 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.

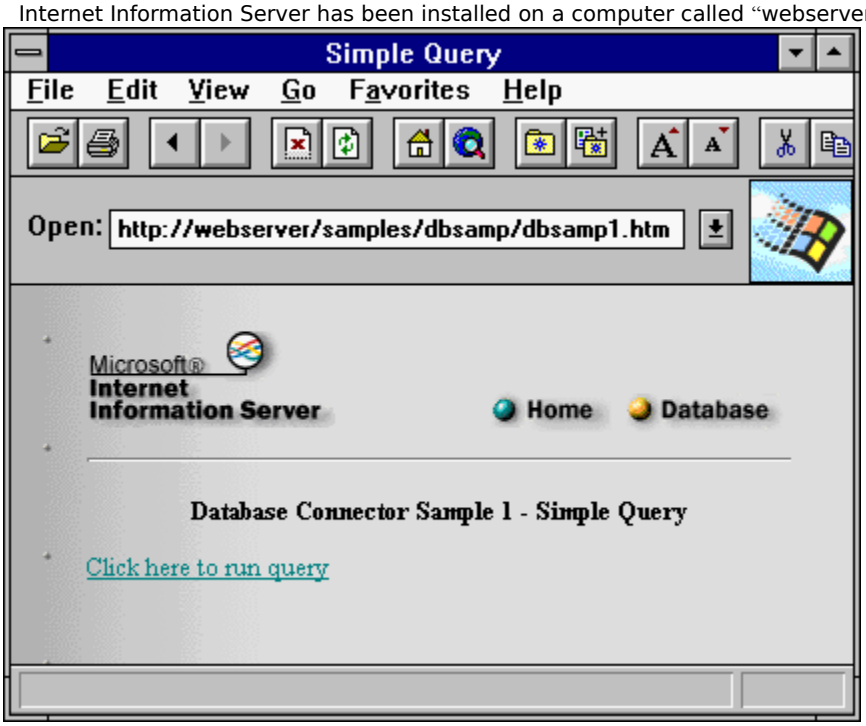
See Walking through a Sample Database Query to see how a Web browser requests information from the Database Connector and how that information is returned.

See the sample HTML pages in the \samples directory for additional tools and information about configuring Microsoft Internet Information Server with ODBC databases.

**Walking through a Sample Database Query**

This example starts with a simple Web page called Sample.htm. The sample Web page will contain one hyperlink that will result in a query being executed using the ODBC driver for Microsoft SQL Server, with the results returned as another Web page.

The following graphic shows Dbsamp1.htm as it is displayed by Microsoft Internet Explorer (assuming that Internet Information Server has been installed on a computer called "webserver").

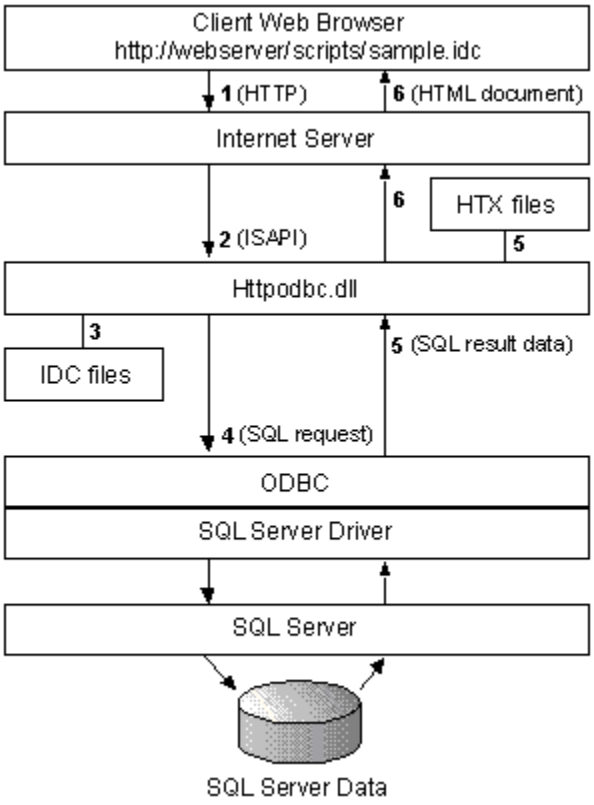



When the hyperlink "Click here to run query" is clicked, another Uniform Resource Locator (URL) is sent to the server. The URL precedes the hyperlink text (it is formatted as hidden text):

```
<A HREF="http://webserver/samples/dbsamp/dbsamp1.idc?">Click here to run query</A>
```

The Internet Database Connector file for Httpodbc.dll to use (Dbsamp1.idc) is referenced in the URL. Extension File Mapping) precludes the need to reference Httpodbc.dll in the URL.

On Internet Information Server, the entire process of using the Internet Database Connector for this example is performed in six steps, as shown in the following diagram.




**Step 1: The URL is received by Internet Information Server.**

The URL is sent by the Web browser.

**Step 2: Internet Information Server loads Httpodbc.dll and provides it with the remaining information in the URL.**

.Idc files are mapped to Httpodbc.dll. Httpodbc.dll loads and obtains the name of the Internet Database Connector file (and other items) from the URL passed to Internet Information Server.

**Step 3: Httpodbc.dll reads the Internet Database Connector file.**

The Internet Database Connector file contains several entries in the format
field: value

In the Sample.idc file, the ODBC data source is specified by:
Datasource: Web SQL

And the HTML extension file is specified by:
Template: sample.htx

Here are the entire contents of the file Sample.idc referenced in the preceding URL:

```
Datasource: Web SQL
Username: sa
Template: sample.htx
SQLStatement:
    +SELECT au_lname, ytd_sales
    + from pubs.dbo.titleview
    + where ytd_sales>5000
```

In the sample .idc file the data source name is "Web SQL". The ODBC installation notes tell how to create a data source called Web SQL.

The other items contained in the sample .idc file include:

- Username, which must be a valid logon to the ODBC datasource; in this example, the logon is to SQL

Server.

- Template, which specifies the file to use to merge the results.
- SQLStatement, which contains the SQL statement to execute.

See Learning the Features of the Internet Database Connector for definitions for all the fields that can be specified in Internet Database Connector files.

The SQLStatement in Sample.idc returns all the author last names and year-to-date sales in units from the "pubs" sample database in SQL Server for authors whose books have year-to-date sales of more than 5000 copies.

**Step 4: Httpodbc.dll connects to the ODBC data source, and executes the SQL statement contained within the Internet Database Connector file.**

The connection is made to the ODBC data source by Httpodbc.dll, which in this example loads the ODBC driver for SQL Server and connects to the server specified in the definition of the data source. Once the connection is made, the SQLStatement in the Internet Database Connector file is sent to the SQL Server ODBC driver, which in turn sends it to SQL Server.

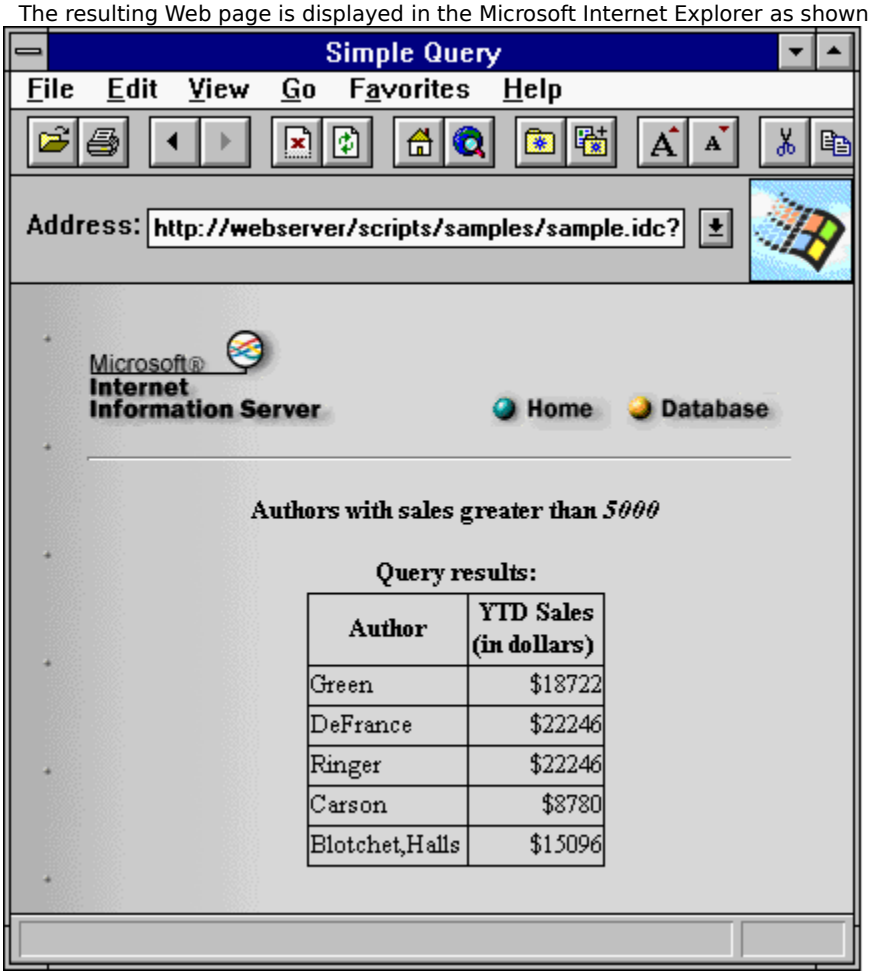
**Step 5: Httpodbc.dll fetches the data from the database, and merges it into the HTML extension file.**

After the SQL statement has been executed, Httpodbc.dll reads the HTML extension file specified in Sample.idc (Sample.htx). HTML extension (.htx) files contains special HTML tags which Httpodbc.dll uses to control where and how the data returned from the SQL statement is merged. Some of these special HTML tags are shown in Sample.htx.

**Step 6: Httpodbc.dll sends the merged document back to Internet Information Server, which returns it to the client.**

After all the data has been merged into Sample.htx, the complete HTML document is sent back to the client.

The resulting Web page is displayed in the Microsoft Internet Explorer as shown following.



**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 Data Source Names (DSNs) 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.

**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 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.

   **Important:** 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 an ODBC driver in the list box and click OK. A dialog box specific to your driver 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 driver and any other information required to access the data, such as the actual server name or location of the database. The data source name is used in Internet Database Connector files to tell Internet Information Server where to access the data.

   For Microsoft SQL Server, the server name, network address, and network library displayed in the Setup dialog box 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.

**Understanding the Sample.htx File**

The Sample.htx file is an HTML document that contains Internet Database Connector tags for data returned from the database (the tags are shown in bold for clarity). Some HTML formatting has been removed to highlight the IDC tags.

[Most of the HTML formatting has been removed for clarity.]

<HTML>

<BODY>

<HEAD><TITLE>Authors and YTD Sales</TITLE></HEAD>

**<%if idc.sales eq ""%>**

<H2>Authors with sales greater than <I>5000</I></H2>

**<%else%>**

<H2>Authors with sales greater than <I><%idc.sales%></I></H2>

**<%endif%>**

<P>

**<%begindetail%>**

**<%if CurrentRecord EQ 0 %>**

Query results:

<B>Author YTD Sales<BR></B>

**<%endif%>**

**<%au_lname%>**$**<%ytd_sales%>**

**<%enddetail%>**

<P>

**<%if CurrentRecord EQ 0 %>**

<I><B>Sorry, no authors had YTD sales greater than </I>**<%idc.sales%>**.</B>

<P>

**<%else%>**

<HR>

<I>

The Web page you see here was created by merging the results

of the SQL query with the template file Sample.htx.

<P>

The merge was done by the Microsoft Internet Database Connector and

the results were returned to this Web browser by the Microsoft Internet Information Server.

</I>

**<%endif%>**

</BODY>

</HTML>

The <%begindetail%> and <%enddetail%> sections delimit where rows returned from the database will appear in the document. Columns returned from the query are surrounded by <%%>, such as <%au_lname%> and <%ytd_sales%> in this example.

**Learning the Features of the Internet Database Connector**

The Internet Database Connector has several features that help create Web pages containing data from a database.

**Internet Database Connector Files**

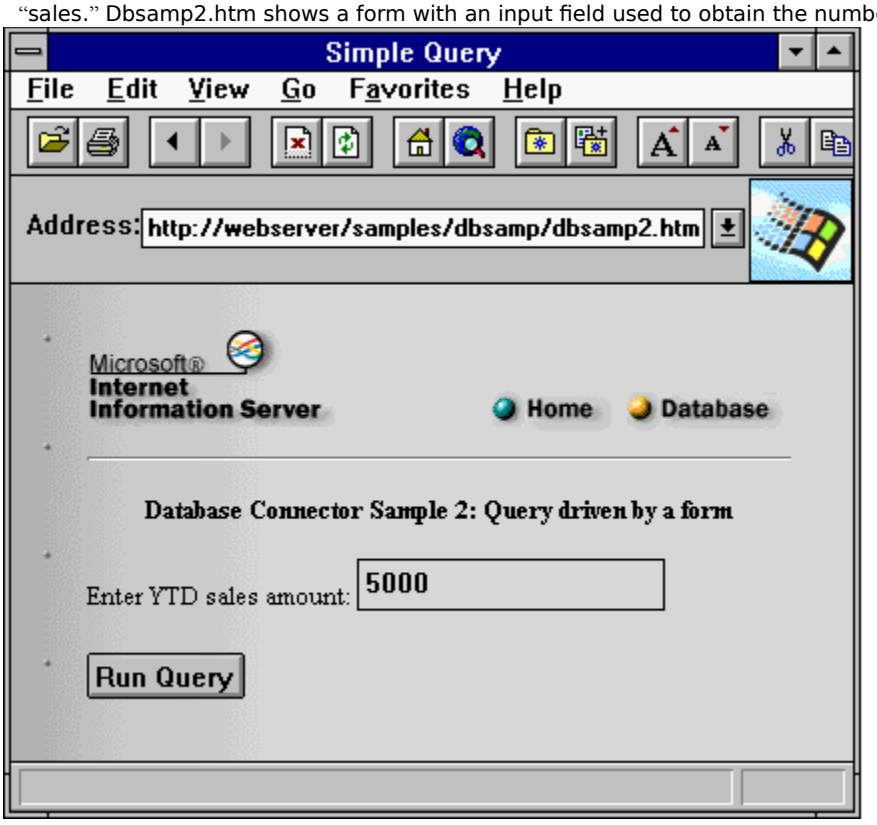
As seen in the examples, Internet Database Connector files contain the information used to access the database. The following section describes the features of Internet Database Connector files.

**Parameters**

The example in the preceding section shows only the simplest kind of query, a query which was defined completely in the Internet Database Connector file. Although this type of query is useful, even more powerful Web pages can be constructed through the use of parameters. Parameters are the names and values of HTML-form controls, such as "<INPUT...>", and names specified directly in URLs. These names and values are sent by Web browsers and can be used in SQL statements on the server.

For example, in the last section the query in Sample.idc returned only the authors whose year-to-date sales exceeded 5000. By using a parameter, you could build a Web page that asks the user to decide what number to use instead of 5000.

The Web page must prompt the user for the year-to-date sales figure and then name the associated variable to "sales." Dbsamp2.htm shows a form with an input field used to obtain the number:




The HTML syntax for the input field and button in Sample2.htm is:

```
<FORM METHOD="POST" ACTION="/scripts/samples/sample2.idc">
<P>
Enter YTD sales amount: <INPUT NAME="sales" VALUE="5000" >
<P>
<INPUT TYPE="SUBMIT" VALUE="Run Query">
</FORM>
```

In the Internet Database Connector file Sample2.idc, you use the parameter shown in bold in place of the number 5000:

```
SQLStatement:
+SELECT au_lname, ytd_sales
+ from pubs.dbo.titleview
+ where ytd_sales > %sales%
```

Here the parameter name must be "sales" so that it corresponds to the <INPUT NAME= "sales" ...> on the Web page. Parameters must be enclosed with percent characters (%) to distinguish them from a normal identifier in SQL. When the Internet Database Connector encounters the parameter in the .idc file, the Internet Database Connector substitutes the value sent by the Web browser and then sends the SQL statement to the ODBC driver.

You can build powerful collections of Web pages by using the output of one query to provide links to other queries. For example, to show the titles for an individual author, instead of returning the author name as plain text, you can format it as a link and then use the link to do another query.

Another example included Internet Information Server shows how to do this type of linkage. Dbsamp3.htm is used to run the query in Sample3.idc, which uses Sample3.htx for the output template. Sample3.htx will return author last names as links, which, when clicked, will display the titles for each author by using Sample3a.idc and Sample3a.htx.

**Fields in Internet Database Connector (.idc) Files**

The following tables list the fields that can be specified in an Internet Database Connector file.

**Required Fields in an Internet Database Connector (.idc) File**

| Field | Description |
|---|---|
| Datasource | The name that corresponds to the ODBC data source name you created earlier by using the ODBC Administrator or the tool provided with the samples. |
| Template | The name of the HTML extension file that formats the data returned from this query. By convention these files use the extension .htx. |
| SQLStatement | The SQL statement to execute. The SQL statement can contain parameter values, which must be enclosed with percent characters (%) from the client. The SQLStatement can occupy multiple lines in the Internet Database Connector file. Following the SQLStatement field, each subsequent line beginning with a plus sign (+) is considered part of the SQLStatement field. |

**Optional Fields in an Internet Database Connector (.idc) File**

| Field | Description |
|---|---|
| DefaultParameters = *param=value* [**,** *param=value*] [...] | The parameter values, if any, that will be used in the Internet Database Connector file if a parameter is not specified by the client. |
| Expires | The number of seconds to wait before refreshing a cached output page. If a subsequent request is identical, the cached page will be returned without ever accessing the database. The Expires field is useful when you want to force a requery of the database after a certain period of time. Httpodbc.dll does not cache output pages. It caches them only when the Expires field is used. |
| MaxFieldSize | The maximum buffer space allocated by Httpodbc.dll per field. Any characters beyond this will be truncated. The parameter applies only to fields returned froms the database that exceed 8192 bytes. The default |

| | |
|---|---|
| | value is 8192 bytes. |
| MaxRecords | The maximum number of records that Httpodbc.dll will return from any one query. The MaxRecords value is not set by default, meaning that a query can return up to 4 billion records. Set this value to limit the records returned. |
| Password | The password that corresponds to the username. If the password is null, this field can be left out. |
| RequiredParameters | The parameter names, if any, that Httpodbc.dll will ensure are passed from the client; otherwise, it will return an error. Parameter names are separated with a comma. |
| Username | A valid user name for the data source name supplied in the Datasource field.<br>**Note:** If you use Microsoft SQL Server with the integrated security option, the username and password fields in the .idc file are ignored. The logon to SQL Server is performed using the credentials of the user. If the request is made as the anonymous user, the username and password are determined by the settings for the anonymous user (IUSR_*computername* by default) in the Internet Service Manager. If the client request contained logon credentials, the username and password supplied by the end user are used to log on to SQL Server. |
| Content Type | Any valid MIME type describing what will be returned to the client. Almost always this will be "text/html" if the .htx file contains HTML. See Server MIME Mapping for more information about adding MIME types. |

**ODBC Advanced Optional Fields**

ODBC advanced options allow debugging and fine tuning of the ODBC driver used by the Internet Database Connector. For more details about these options, consult your ODBC driver documentation or the ODBC Software Developer's Kit. The format in the IDC file is:

ODBCOptions: Option Name=Value[,Option Name=Value...]

For example, to stop the SQL statement from running for more than 10 seconds and enabling tracing of ODBC function calls, in the IDC file you would specify:

ODBCOptions: SQL_QUERY_TIMEOUT=10, SQL_OPT_TRACE=1

All options are described in the following table:

| Option Name | Value | Purpose |
|---|---|---|
| SQL_ACCESS_MODE | 0 = Read/Write<br>1 = Read Only. | An indicator for the ODBC driver or data source that the connection is not required to support SQL statements that cause updates to occur. This mode can be used to optimize locking strategies, transaction management, or other areas as appropriate to the driver or data source. The driver is not required to prevent such statements from being submitted to the data source. The behavior of the driver and data source when asked to process SQL statements that are not read-only during a read-only connection is implementation-defined. SQL_ACCESS_MODE set to 0 is the default, which allows reading and writing . |
| SQL_LOGIN_TIMEOUT | Integer | The number of seconds to wait for a log on request to complete before aborting the connection. The default is driver-dependent and must be nonzero. If the value is 0, the timeout is |

| | | |
|---|---|---|
| | | disabled and a connection attempt will wait indefinitely. If the specified timeout exceeds the maximum log on timeout in the data source, the driver substitutes that value. |
| SQL_OPT_TRACE | 0 = Trace off<br>1 = Trace on | When tracing is on, each ODBC function call made by Httpodbc.dll is written to the trace file. You can specifiy a trace file with the SQL_OPT_TRACEFILE option. If the file already exists, the ODBC appends to the file. Otherwise, it creates the file. If tracing is on and no trace file has been specified, ODBC writes to the file \ SQL.LOG. |
| SQL_OPT_TRACEFILE | Filename | The name of the trace file to use when SQL_OPT_TRACE=1. The default is \SQL.LOG |
| SQL_PACKET_SIZE | Integer | The network packet size, in bytes, to be used to exchange information between the DBMS and the Web Server.<br>**Note** Many data sources either do not support this option or can return only the network packet size. If the specified size exceeds the maximum packet size or is smaller than the minimum packet size, the driver substitutes that value. |
| SQL_TRANSLATE_DLL | Filename | The name of a DLL containing the functions SQLDriverToDataSource and SQLDataSourceToDriver that the driver loads and uses to perform tasks such as character set translation. |
| SQL_TRANSLATE_OPTION | Integer | Value controlling translation functionality, which is specific to the Translation DLL being used. Consult the documentation for the driver and translation DLL for details. |
| SQL_TXN_ISOLATION | Integer<br>1=Read Uncommited<br>2=Read Committed<br>4=Repeatable Read<br>8=Serializable<br>16=Versioning | Sets the transaction isolation level. The Internet Database Connector does not support transactions than span more than the request in the IDC file. However, for some DBMSs, setting the SQL_TXN_ISOLATION option to 1 (Read Uncommited) will result in higher concurrency and therefore better performance. However, with this setting, data may be retrieved which has not been committed to the database by other transactions. |
| SQL_MAX_LENGTH | Integer | The maximum amount of data that the driver returns from a character or binary column. This option is intended to reduce network traffic and should only be used when the data source (as opposed to the driver) in a multiple-tier driver can implement it. |
| SQL_MAX_ROWS | Integer | The maximum number of rows to return for a SELECT statement. If the value equals 0 (the default), then the driver returns all rows.This option is intended to reduce network traffic when |

| | | |
|---|---|---|
| | | the data source itself can limit the return rows, as opposed to the MaxRecords built-in variable in the Internet Database Connector, which limits the rows fetched. |
| SQL_NOSCAN | 0=Scan for and convert escape clauses<br><br>1=Do not scan for and convert escape clauses | Specifies whether the driver does not scan SQL strings for escape clauses. If set to 0 (the default), the driver scans SQL strings for escape clauses. If set to 1, the driver does not scan SQL strings for escape clauses; instead, the driver sends the statement directly to the data source. If your SQL statement does not contain any ODBC escape clauses, a special syntax enclosed by curly braces ( { } ), then setting this option to 1 will provide a small performance gain by directing the driver to not scan the SQL string. |
| SQL_QUERY_TIMEOUT | Integer<br><br>0=No timeout | The number of seconds to wait for a SQL statement to execute before aborting the query. When set to 0 (the default) there is no timeout. If the specified timeout exceeds the maximum timeout in the data source, or is smaller than the minimum timeout, the driver substitutes that value. |
| Integer | Driver Specific | Driver-specific option values can be specified in the form *number=value*. For example,<br>`4322=1, 234=String` |

**Using Select Multiple List Boxes in HTML Forms**

When an HTML form containing a <SELECT MULTIPLE ...> tag is used, the Internet Database Connector converts the items selected into a comma-separated list; the list can be used in the .idc file just like other parameters. However, because the parameter is actually a list, it will typically only be used for SQLSelect statements with an IN clause, as in the following examples.

If the parameter name in the .idc file is enclosed in single quotation marks, each element of the list will be enclosed in single quotation marks also. You should enclose the parameter name in single quotation marks whenever the column in the IN clause is a character column or other type in which literals are quoted (dates and times, for example). If there are no single quotation marks around the parameter name, no quote will be placed around each element of the list. You should not enclose the parameter name in single quotation marks when the column in the IN clause is a numeric type or any other type in which literals are not enclosed in single quotation marks.

For example, if an HTML form contained the multiple select list box shown below:

```
<SELECT MULTIPLE NAME="region">
<OPTION VALUE="Western">
<OPTION VALUE="Eastern">
<OPTION VALUE="Northern">
<OPTION VALUE="Southern">
</SELECT>
```

You can construct an .idc file with a SQL Statement:

SQLStatement: SELECT name, region FROM customer WHERE region IN ('%region%')

If the user selected "Northern" and "Western" from the HTML form, the SQL statement would be converted to:

SELECT name, region FROM customer WHERE region IN ('Northern', 'Western')

If the user selected "Northern", "Western", and "Eastern" from the HTML form, the SQL statement would be converted to:

SELECT name, region FROM customer WHERE region IN ('Northern', 'Western', 'Eastern')

Another example of an HTML form is shown below, but this time uses numeric data and therefore no quotation marks enclose the parameter in the .idc file.

```
<SELECT MULTIPLE NAME="year">
<OPTION VALUE="1994">
```

<OPTION VALUE="1995">
<OPTION VALUE="1996">
</SELECT>

You can construct an .idc file with an SQLStatement:

SQLStatement: SELECT product, sales_year FROM sales WHERE sales_year IN (%year%)

If the user selected "1994" and "1995" from the HTML form, the SQL statement would be converted to:

SELECT product, sales_year FROM sales WHERE sales_year IN (1994, 1995)

**HTML Extension (.htx) Files**

HTML extension files contain a number of keywords that control how the output HTML document is constructed. These keywords are explained in the following sections.

**<%begindetail%>, <%enddetail%>**

The <%begindetail%>, <%enddetail%> keywords surround a section of the HTML extension file in which the data output from the database will be merged. Within the section, the column names delimited with <% and %> are used to mark the position of the returned data from the query. For example:

```
<%begindetail%>
<%au_lname%>: <%ytd_sales%>
<%enddetail%>
```

will list the columns au_lname and ytd_sales. Any column can be referred to in this way. Column names can also be referred to elsewhere in the HTML extension file. Note: if there are no records returned from the query, the <%begindetail%> section will be skipped.

**<%if%>..<%else%>..<%endif%>**

HTML extension files can contain conditional logic with an if-then-else statement to control how the Web page is constructed. For example, one common usage is to insert a condition to display the results from the query on the first row within a <%begindetail%> section; but if there are no records returned by the query, to display the text "Sorry, no authors had YTD sales greater than" **%idc.sales%**. By using the <%if%> statement and a built-in variable called "CurrentRecord" you can tailor the output so that the error message is printed when no records are returned. Here is an example showing the use of the <%if%> statement.

**<%begindetail%>**

**<%if CurrentRecord EQ 0 %>**

Query results:

<B>Author YTD Sales<BR></B>

**<%endif%>**

**<%au_lname%>**$**<%ytd_sales%>**

**<%enddetail%>**

<P>

**<%if CurrentRecord EQ 0 %>**

<I><B>Sorry, no authors had YTD sales greater than </I>**<%idc.sales%>**.</B>

<P>

**<%else%>**

<HR>

<I>

The Web page you see here was created by merging the results

of the SQL query with the template file Sample.htx.

<P>

The merge was done by the Microsoft Internet Database Connector and

the results were returned to this Web browser by the Microsoft Internet Information Server.

</I>

**<%endif%>**

</BODY>

</HTML>

The general syntax is:

<%if *condition* %>

*HTML text*

[<%else%>

*HTML text*]

<%endif%>

Where *condition* is of the form:

*value1 operator value2*

and *operator* can be one of the following:

| | |
|---|---|
| EQ | if *value1* equals *value2* |
| LT | if *value1* is less than *value2* |
| GT | if *value1* is greater than *value2* |
| CONTAINS | if any part of *value1* contains the string *value2* |

The operands *value1* and *value2* can be column names, one of the built-in variables (CurrentRecord or MaxRecords, see below), an HTTP variable name (see below), or a constant. When used in an <%if%> statement, values are not delimited with <% and %>. For example, to do special processing on author name "Green," use the condition:

```
<%begindetail%>
<%if au_lname EQ "Green"%>
this guy is green!
<%endif%>
<%enddetail%>
```

The <%if%> statement can also be used to do special processing based on information from HTTP variables. For example, to format a page differently based on the type of client Web browser you could include the following in the HTML extension file.

```
<%if HTTP_USER_AGENT contains "Mozilla"%>
client supports advanced HTML features
<%else%>
client is <%HTTP_USER_AGENT%>
<%endif%>
```

**CurrentRecord, MaxRecords**

The CurrentRecord built-in variable contains the number of times the <%begindetail%> section has been processed. The first time through the <%begindetail%> section, the value is zero. Subsequently, the value of CurrentRecord changes every time another record is fetched from the database.

The MaxRecords built-in variable contains the value of the MaxRecords field in the Internet Database Connector file. MaxRecords and CurrentRecord can only be used in <%if%> statements.

**Parameters from Internet Database Connector files**

Parameters from Internet Database Connector files can be accessed in the HTML extension file by prefixing the name of the parameter with "idc" and a period. In Sample3.htx shown earlier, you could show the value of the parameter %sales% by including the line:

```
The value of the sales parameter is: <%idc.sales%>
```

**HTTP variables**

Several variables in HTML extension files can give a lot of information about the environment and Web client connected to the server. In addition all headers sent by the client are available. To access them by using the Internet Database Connector you must convert them:

1. Add HTTP_ to the beginning.

2. Convert all dashes to underscores.

3. Convert all letters to uppercase.

The following table gives a listing of default variables.

| HTTP variable | Meaning |
|---|---|
| ALL_HTTP | All HTTP headers that were not already parsed into one of the listed variables. These variables are of the form HTTP_<header field name>, for example:<br>HTTP_ACCEPT: */*, q=0.300, audio/x-aiff, audio/basic, image/jpeg, image/gif, text/plain, text/html<br>HTTP_USER_AGENT: Microsoft Internet Explorer/0.1 (Win32)<br>HTTP_REFERER: http://webserver/samples/dbsamp/dbsamp3.htm<br>HTTP_CONTENT_TYPE: application/x-www-form-urlencoded<br>HTTP_CONTENT_LENGTH: 10<br>HTTP_EXTENSION: Security/Digest |
| AUTH_TYPE | The type of authorization in use. If the username has been authenticated by the server, this will contain Basic. Otherwise, it will not be present. |
| CONTENT_LENGTH | The number of bytes that the script can expect to receive from the client. |
| CONTENT_TYPE | The content type of the information supplied in the body of a POST request. |
| GATEWAY_INTERFACE | The revision of the CGI (Common Gateway Interface) specification with which this server complies. The current version is CGI/1.1. |
| HTTP_ACCEPT | Special-case HTTP header. Values of the Accept: fields are concatenated, separated by “, ”; for example, if the following lines are part of the HTTP header:<br>accept: */*; q=0.1<br>accept: text/html<br>accept: image/jpeg<br>then the HTTP_ACCEPT variable will have a value of:<br>*/*; q=0.1, text/html, image/jpeg |
| PATH_INFO | Additional path information, as given by the client. This comprises the trailing part of the URL after the script name but before the query string (if any). |
| PATH_TRANSLATED | This is the value of PATH_INFO, but with any virtual path name expanded into a directory specification. |
| QUERY_STRING | The information which follows the question mark (?) in the URL that referenced this script. |
| REMOTE_ADDR | The IP address of the client. |
| REMOTE_HOST | The hostname of the client. |
| REMOTE_USER | This contains the username supplied by the client and authenticated by the server. String . |
| REQUEST_METHOD | The HTTP request method |

| | |
|---|---|
| SCRIPT_NAME | The name of the script program being executed. |
| SERVER_NAME | The server's hostname (or IP address) as it should appear in self-referencing URLs. |
| SERVER_PORT | The TCP/IP port on which the request was received. |
| SERVER_PROTOCOL | The name and version of the information-retrieval protocol relating to this request, usually HTTP/1.0. |
| SERVER_SOFTWARE | The name and version of the Web server under which the Internet Server Extension is running. |

### Using Other Windows NT Tools

In addition to Internet Service Manager you can use other tools to configure, control, and monitor the Internet Information services. This topic explains other Windows NT utilities that directly affect your Internet Information Server, and explains how you can use other Windows NT utilities to monitor or enhance your Internet Information Server site.

### Configuring Server Options with Control Panel

Use Control Panel to set Windows NT–controlled systems and options.

The Network Applet

The Network applet in Control Panel configures your Transmission Control Protocol/Internet Protocol (TCP/IP) settings, including IP address, subnet mask, and default gateway. Double-click TCP/IP Protocol in the Installed Network Software listing to display the TCP/IP Configuration dialog box.

Click the advanced button to set Domain Name System (DNS) settings, such as hostname, domain names, and DNS servers, to resolve names.

The Services Applet

The Services applet is used to start, stop, and pause the WWW, Gopher, and FTP services. You can also use Internet Service Manager to start, stop and pause the services.

Use the Startup button to configure how the service starts when the computer starts. If you have a specific reason, you can also use this dialog box to override the account used by the WWW service as set in the Service property sheet of Internet Service Manager. You should change this setting only if it is part of your security strategy; otherwise, use the default settings in the Log On As box.

The ODBC Applet

The ODBC applet in Control Panel is used to set up ODBC connectivity. See Chapter 8, “Publishing Information and Applications,” for more information about using the ODBC applet.

### Setting File Access with File Manager

Use File Manager to set directory and file permissions on Windows NT File System (NTFS) drives. Use the Permissions item in the Security menu to set permissions.

### Setting User Access with User Manager for Domains

User Manager for Domains, in the Administrative Tools program group, is a tool that you can use to manage security for a Windows NT Server computer. With User Manager you can:

- Create and manage user accounts.
- Create and manage groups.
- Manage the security policies.

### Tracking Problems with Event Viewer

Event Viewer, in the Administrative Tools program group, is a tool that you can use to monitor events in your system. You can use Event Viewer to view and manage System, Security, and Application event logs. Event Viewer can notify administrators of critical events by displaying pop-up messages, or by adding event information to log files. The information allows you to better understand the sequence and types of events that led up to a particular state or situation.

### Monitoring Your Server with Performance Monitor

Windows NT Performance Monitor counters are automatically installed. You can use the Windows NT Performance Monitor for real-time measurement of your service use.

### 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. By default, most MIME types are set up in Internet Information Server. If MIME mapping on the server is not set up for your specific file type, browsers may not be able to retrieve the file. Over 100 common MIME mappings are installed by default.

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 directories.

**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**

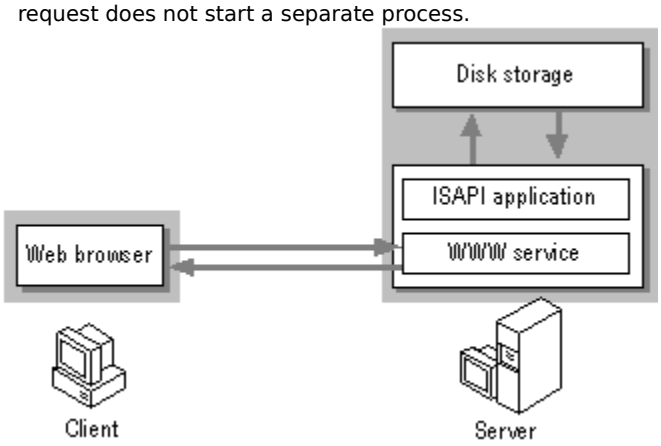
Interactive applications or scripts can be written in almost any 32-bit programming language, such as C or Perl, or as Windows NT batch files (.bat or .cmd). When you write your applications or script you can use one of two supported interfaces, the Microsoft Internet Server Application Programming Interface (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.

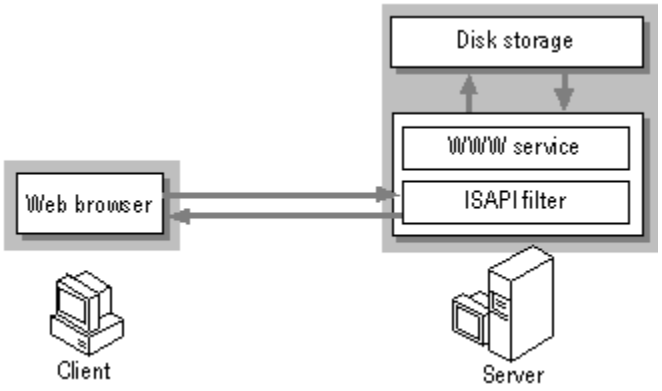
**Internet Server API**

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.

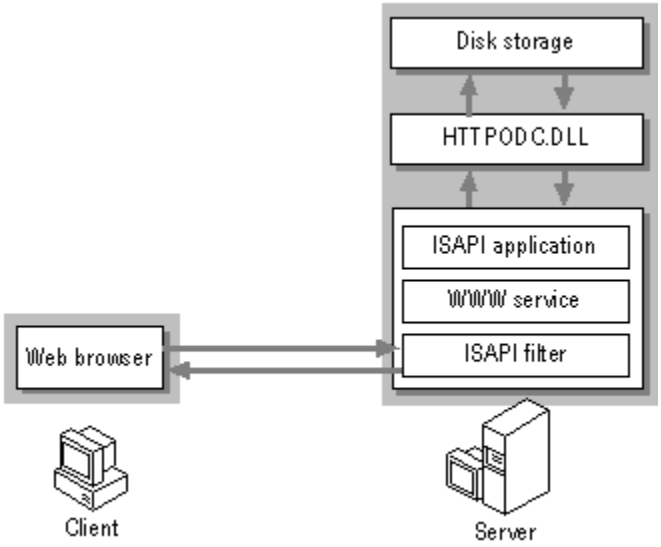
ISAPI can be used to create applications that run as DLLs on your Web server. If you have used Common Gateway Interface (CGI) scripts before, you will find that the ISAPI applications have much better performance because your applications are loaded into memory at server run-time. They require less overhead 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.

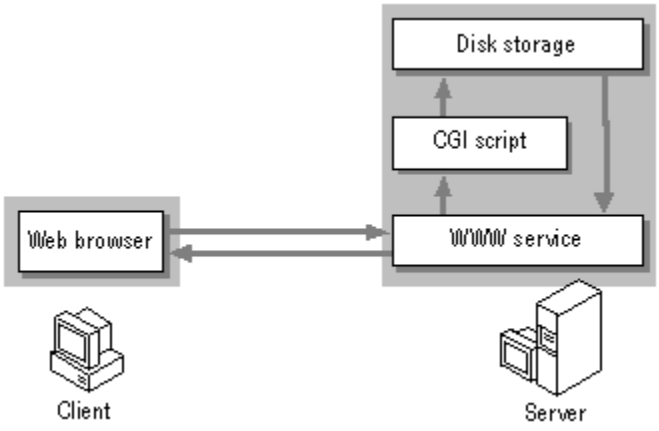



For complete information about programming with ISAPI, see the Microsoft BackOffice Software Developer's Kit (SDK), available from Microsoft. See the introductory chapter, "Before You Begin," in the Microsoft Internet Information Server *Installation and Planning Guide* for further information about 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 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.

| Extension | Default Interpreter |
|---|---|
| .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.

**Securing Your Site Against Intruders**

Connecting computers to the Internet provides for some very powerful and useful scenarios. Within hours it becomes possible to communicate with millions of people and computers worldwide using Transfer Control Protocol/Internet Protocol (TCP/IP). This broad flexibility imposes a degree of risk — not only can you communicate with people and other systems, it is also possible for users to attempt to initiate communication with your system. Although connecting to servers on the Internet is generally done with good intentions, there are malicious individuals who attempt to infiltrate internal networks. The Windows NT operating system was designed with security in mind.

Your security configuration is crucial for safe operation of your server on the Internet. Although it is unlikely that your site will be maliciously tampered with, Internet servers are available to the general public and there may be some degree of public intrusion. You should understand all of the security implications of connecting your computer to a public network.

Securely Configuring Windows NT Server

Authentication and Security Integration with Windows NT

Anonymous Connections

Client Requests Containing Credentials

Internet Service Manager Authentication Options

Other Authentication Issues

How Internet Information Server Security Works

Securely Configuring the WWW Service

Securing Data Transmissions with Secure Sockets Layer (SSL)

If you do not understand the implications of the security topics, you should consult Windows NT documentation, an authorized Microsoft Solution Provider, or other source before installing your site on the Internet.

**Securely Configuring Windows NT Server**

Windows NT provides user-account security and Windows NT File System (NTFS) file-system security. You can use the topics below as a checklist to ensure you have effectively used User Accounts and NTFS to secure Windows NT Server. Additionally, you can prevent security breaches by properly configuring the services running on your computer.

**Preventing Intrusion by Setting Up User Accounts**

Windows NT security helps you protect your computer and its resources by requiring assigned user accounts. You can control access to all computer resources by limiting the user rights of these accounts.

Every operation on a computer running Windows NT identifies who is doing the operation. For example, the username and password that you use to log on to Windows NT identifies who you are and defines what you are authorized to do on that computer.

What a user is authorized to do on a computer is configured in User Manager by setting User Rights in the Policies menu. User rights authorize a user to perform certain actions on the system, including the right to "Log on locally," which is required for users to use Internet Information Server services.

**Allowing Anonymous Access with the IUSR_*computername* Account**

The IUSR_*computername* account is created during Microsoft Internet Information Server setup. For example, if the computer name is marketing1, then the anonymous access account name is IUSR_marketing1.

By default, all Microsoft Internet Information Server client requests use this account. In other words, information server clients are logged on to the computer using the IUSR_*computername* account. The IUSR_*computername* account is permitted only to log on locally. No network rights are granted that could allow an unauthorized user to damage your server or its files.

**Note:** The IUSR_*computername* account is also added to the group Guests. If you have changed the settings for the Guests group, those changes also apply to the IUSR_*computername* account. You should review the settings for the Guests group to ensure that they are appropriate for the IUSR_*computername* account.

If you allow remote access only by the IUSR_*computername* account, remote users do not provide a username and password, and have only the permissions assigned to that account. This prevents hackers from attempting to gain access to sensitive information with fradulent or illegally obtained passwords. For some situations this can provide the best security.

**Requiring a Username and Password**

Conversely, if you require "authenticated" clients, users must supply a valid Windows NT username and password.

Basic authentication does not encrypt your username and password before transmission. Basic authentication is encoded only by using UUencode, and can be decoded easily by anyone with access to your network, or to a segment of the Internet that transfers your packets.

**Warning:** Using basic authentication means you that will send your Windows NT username and password unencrypted over public networks. Intruders could easily learn usernames and passwords.

The WWW service also supports the Windows NT Challenge/Response encrypted password transmission. Microsoft recommends only the Windows NT Challenge/Response method of password authentication.

Windows NT authentication, currently supported only by Microsoft Internet Explorer for Windows 95, encrypts the username and password, providing secure transmission of usernames and passwords over the Internet.

With both basic authentication and Windows NT authentication, no access is permitted unless a valid username and password is supplied. Password authentication is useful if you want only authorized individuals to use your server or specific portions controlled by NTFS. You can have both IUSR_*computername* access and authenticated access enabled at the same time.

**Choose Difficult Passwords**

The easiest way for someone to gain unauthorized access to your system is with a stolen or easily guessed

password. Make sure that all passwords used on the system, especially those with administrative rights, have difficult-to-guess passwords. In particular make sure to select a good administrator password (a long, mixed-case, alphanumeric password is best) and set the appropriate account policies. Passwords can be set by using the User Manager utility, or at the system logon prompt.

**Maintain Strict Account Policies**

The User Manager utility provides a way for the system administrator to specify how quickly account passwords expire (which forces users to regularly change passwords), and other policies such as how many bad logon attempts will be tolerated before locking a user out. Use these policies to manage your accounts, particularly those with administrative access, to prevent exhaustive or random password attacks.

**Limit the Membership of the Administrator Group**

By limiting the members of the Administrator group, you limit the number of users who might choose bad passwords and expose your system.

**NTFS File Security**

You should place your data files on an NTFS partition to provide additional security and access control. You can limit access to portions of your file system for specific users and services by using NTFS. In particular, it is a good idea to apply Access Control Lists (ACLs) to your data files for any Internet publishing service.

The NTFS file system gives you very granular control on files by specifying users and groups that are permitted access and what type of access they may have for specific files and directories. For example, some users may have Read-only access, while others may have Read, Change, and Write access. You should ensure that the IUSR_*computername* or authenticated accounts are granted or denied appropriate access to specific resources.

You should note that the group Everyone contains all users and groups, including the IUSR_*computername* account and the Guests group. By default the group Everyone has full control of all files created on an NTFS drive.

If there are conflicts between your NTFS settings and Microsoft Internet Information Server settings, the strictest setttings will be used.

You should review the security settings for all Microsoft Internet Information Server directories and adjust them appropriately. Generally you should use the settings in the following table:

| **Directory Type** | **Suggested Access** |
|---|---|
| content | Read access |
| programs | Read and Execute access |
| databases | Read and Write access |

**Enable Auditing**

You can enable auditing of NTFS files and directories on Windows NT Server through the File Manager. You can review the audit records periodically to ensure that no one has gained unauthorized access to sensitive files.

**Running Other Network Services**

You should review all of the network services that you are using on any computer connected to the Internet.

**Run Only the Services that You Need**

The fewer services you are running on your system, the less likely a mistake will be made in administration that could be exploited. Use the Services applet in the Windows NT Control Panel to disable any services not absolutely necessary on your Internet server.

**Unbind Unnecessary Services from Your Internet Adapter Cards**

Use the Bindings feature in the Network applet in the Windows NT Control Panel to unbind any unnecessary services from any network adapter cards connected to the Internet. For example, you might use the Server service to copy new images and documents from computers in your internal network, but you might not want

remote users to have direct access to the Server service from the Internet. If you need to use the Server service on your private network, the Server service binding to any network adapter cards connected to the Internet should be disabled. You can use the Windows NT Server service over the Internet; however, you should fully understand the security implications and licensing issues.

The FTP Server service included with Windows NT should also be disabled or configured to ensure adequate security. You must remove the standard FTP Server service in Windows NT if the Microsoft Internet Information Server FTP service will be installed.

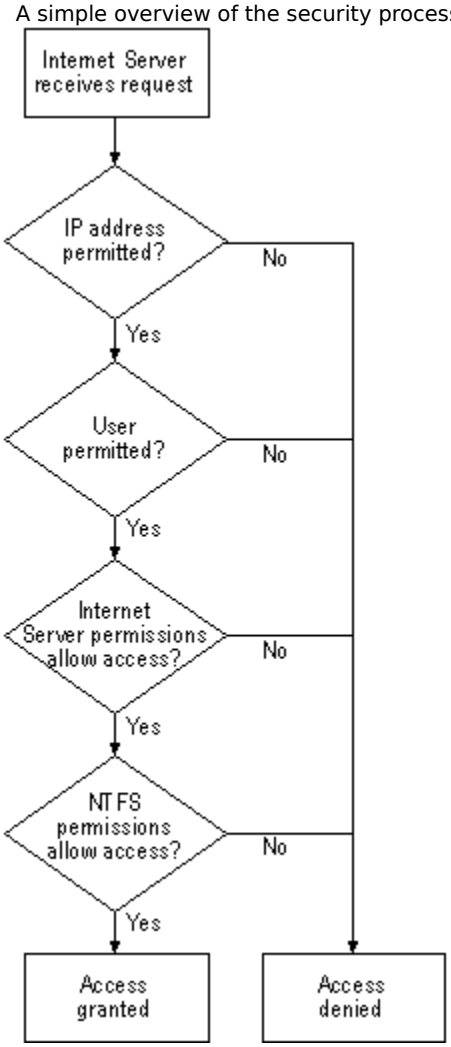
**Check Permissions on Network Shares**

If you *are* running the Server service on your Internet adapter cards, be sure to double-check the permissions set on the shares you have created on the system. It is also wise to double-check the permissions set on the files contained in the shares' directories to ensure that you have set them correctly.

**How Internet Information Server Security Works**

Internet Information Server integrates Windows NT authentication (username and password) security and NTFS file system security. Additional security is implemented by the Internet Information Server by using IP address security and directory access settings.

A simple overview of the security process used on each request is presented in the following illustration.




**IP Address Security**

The source IP address of every packet received is checked against the Internet Information Server settings in the Advanced property sheet. If Internet Information Server is configured to allow access by all computers except those listed as exceptions to that rule, access is denied to any computer with an IP address included in that list. Conversely, if Internet Information Server is configured to deny all IP addresses, access is denied to all remote users except those whose IP addresses have been specifically granted access.

IP address security is probably most useful on the Internet to exclude everyone except known users. IP address security can also be used to exclude individuals or entire networks that you do not want to grant access to.

**Username Authentication**

By default, all requests use anonymous access through the IUSR_*computername* user account created during Internet Information Server setup. This account is a user account and it granted the right to log on locally.

Username authentication is probably most useful if you want to control access to your server by individual user or group.

### Internet Server Permissions

When you assign home and virtual directories for use by Internet Server services, you also specify the type of access that users have for the files in that directory. The WWW service allows you to assign these permissions to a directory:

**Read**

Allows users to view files contained in a directory.

**Execute**

Allows users to start applications or scripts. All Internet Server API (ISAPI) applications and Common Gateway (CGI) scripts must be placed into the default \Scripts directory or into a directory configured with Execute permission.

Execute pemission must be set in both Internet Service Manager and File Manager when you install applications and scripts on an NTFS drive.

Install server applications into a directory that is configured for Execute in Internet Service Manager and is *not* configured for Write permission on an NTFS drive. This will prevent malicious users from copying programs to your computer that could damage it when run.

**Require Secure SSL Channel**

Allows users to send information to the server in encrypted format, ensuring data privacy.

Note that the FTP service supports Read and Write only; the Gopher service supports Read only.

### NTFS Permissions

On NTFS drives you must also ensure that similar permissions are set on directories.

### Securely Configuring the WWW Service

In addition to implementing the previous suggestions on securing Windows NT Server, you can further enhance your security by using Internet Service Manager to configure the WWW service.

#### Controlling Access by Username

To gain access to files, users must be identified with a valid username and password.

If you are allowing anonymous logon using the IUSR_*computername* account, you should first ensure that computer-wide User Rights (in the User Manager Policies menu) do not allow the IUSR_*computername* account, the Guests group, or the Everyone group any right other than to "Log on Locally." Next, ensure that the file permissions set in the Windows NT File Manager are appropriate for all content directories used by Microsoft Internet Information Server.

If you allow basic or Windows NT Challenge/Response password authentication, users can supply a username and password to gain access to areas that require specific authorization. The usernames must be valid usernames on the computer running Internet Information Server, or in a Windows NT domain accessible from that computer.

#### Controlling Access by IP Address

Microsoft Internet Information Server can be configured to grant or deny access to specific IP addresses. For example, you can exclude a harassing individual by denying access to your server from a particular IP address, or prevent entire networks from accessing your server. Conversely, you can choose to enable only specific sites to have access to your service.

Use the Advanced property sheet for the appropriate information service to limit access by IP address.

Choose the Granted Access button or the Denied Access button, and then list the exceptions by using the Add button.

Choose Single Computer and provide the IP address to exclude a single computer. Choose Group of Computers and provide an IP address and subnet mask to exclude a group of computers.

You are specifying (by IP address) which computer or group of computers will be granted or denied access. If you choose to grant access to all computers by default, you can then specify the computers to be denied access. Conversely, if you choose to deny access to all users by default, you can then specify which computers are allowed access.

#### Disable Directory Browsing

Unless it is part of your strategy, you should disable directory browsing on the Directories property sheet. Directory browsing exposes the entire file structure; if it is not configured correctly, you run the risk of exposing program files or other files to unauthorized access.

**Securing Data Transmissions with Secure Sockets Layer (SSL)**

Microsoft Internet Information Server offers a protocol for providing data security layered between its service protocols (HTTP) and TCP/IP. This security protocol, called Secure Sockets Layer (SSL), provides data encryption, server authentication, and message integrity for a TCP/IP connection.

**About SSL**

SSL is a protocol submitted to the W3C working group on security for consideration as a standard security approach for World Wide Web browsers and servers on the Internet.

SSL provides a security "handshake" that is used to initiate the TCP/IP connection. This handshake results in the client and server agreeing on the level of security that they will use and fulfills any authentication requirements for the connection. Thereafter, SSL's only role is to encrypt and decrypt the byte stream of the application protocol being used (for example, HTTP). This means that all the information in both the HTTP request and the HTTP response are fully encrypted, including the URL the client is requesting, any submitted form contents (such as credit card numbers), any HTTP access authorization information (usernames and passwords), and all the data returned from the server to the client.

An SSL-enabled server can send and receive private communication across the Internet to SSL-enabled clients (browsers), such as Microsoft Internet Explorer version 2.0 for Windows 95 (included on the Microsoft Internet Information Server compact disc in the \Clients directory).

Enabling SSL security on a Microsoft Internet Information Server involves the following steps:

1. Generate a key pair file and a request file.
2. Request a certificate from a certification authority.
3. Install the certificate on your server.
4. Activate SSL security on a WWW service directory.

**Important:** Keep in mind the following points when enabling SSL security:

- You can enable SSL security on the root of your Web home directory (\Wwwroot by default) or on one or more virtual directories.
- Once enabled and properly configured, only SSL-enabled clients will be able to communicate with the SSL-enabled WWW directories.
- URLs that point to documents on a SSL-enabled WWW directory must use "https://" instead of "http://" in the URL. Any links using "http://" in the URL will not work on a secure directory.
- SSL security is enabled and disabled by using Internet Service Manager.

**Generating a Key Pair**

Use the Keygen utility to create two files. (Keygen.exe is in C:\Inetsrv\Server or the directory where you installed Microsoft Internet Information Server.) The first file is a key file containing the key pair; the second file is a certificate request file (type **keygen** with no arguments for command syntax and an example).

Before starting, you must have decided on the server's Distinguished Name attributes. The Distinguished Name attributes are enclosed in quotation marks in the example below.

The following example creates the key file keypair.key and the certificate request file named Request.req for a server named www.mycompany.com: The files are generated in the current directory, C:\Inetsrv\Server.

```
c:\inetsrv\server>keygen MyPassword1 keypair.key request.req
"C=US,S=WASHINGTON,L=REDMOND,O=EXAMPLE,OU=TOUR,CN=www.mycompany.com"

    SSL Key generation utility, Version 1.0

    Copyright (c) 1995 Microsoft Corporation

    Generating key pair of length 1024 bits...

    Completed.

    Send the generated request file, Request.req,

    to your Certificate Authority for signing.
```

The argument in quotes in the Keygen.exe command line ("C=US, S=Washington...") specifies several fields for

the Distinguished Name used in the certificate request related to your organization and server.

The Distinguished Name attributes for Keygen.exe follow:

C= 2 Letter ISO Country designations (for example, US, FR, AU, UK, DE)

S= State or Province (for example, Washington, Alberta, or California — do not abbreviate)

L= Locality (for example, Redmond, Calgary, or Redwood City)

O= Organization (Preferably ISO-registered top-level organization or company name)

OU= Organizational Unit

CN= Common Name (Domain Name of server, for example, www.mycompany.com).

**Note:** Do not use commas in any field. Commas are interpreted as the end of that field and will generate an invalid request without warning.

If you run Keygen.exe more than once, note that it does not overwrite existing files; instead, it returns an error 80, meaning that the file already exists.

### Acquiring a Certificate

In order to use SSL, you will need a certificate from a certifying authority such as VeriSign for your system. Instructions for acquiring a VeriSign certificate can be found on VeriSign's Web site, www.verisign.com.

### Applying Your Certificate to Your Server

After you complete your certificate request, you will receive a signed certificate from the certification authority (consult your certification authority for complete details). It will look something like the following example:

```
-----BEGIN CERTIFICATE-----

 JIEBSDSCEXoCHQEwLQMJSoZILvoNVQECSQAwcSETMRkOAMUTBhMuVrMmIoAnBdNV
 BAoTF1JTQSBEYXRhIFNlY3VyaXR5LCBJbmMuMRwwGgYDVQQLExNQZXJzb25hIENl
 cnRpZmljYXRlMSQwIgYDVQQDExtPcGVuIE1hcmtldCBUZXN0IFNlcnZlciAxMTAw
 HhcNOTUwNzE5MjAyNzMwWhcNOTYwNTE0MjAyOTEwWjBzMQswCQYDVQQGEwJVUzEg
 MB4GA1UEChMXUlNBIERhdGEgU2VjdXJpdHksIEluYy4xHDAaBgNVBAsTE1BlcnNv
 bmEgQ2VydGlmaWNhdGUxJDAiBgNVBAMTG09wZW4gTWFya2V0IFRlc3QgU2VydmVy
 IDExMDBcMA0GCSqGSIb3DQEBAQUAA0sAMEgCQQDU/7lrgR6vkVNX40BAq1poGdSm
 GkD1iN3sEPfSTGxNJXY58XH3JoZ4nrF7mIfvpghNi1taYimvhbBPNqYe4yLPAgMB
 AAEwDQYJKoZIhvcNAQECBQADQQBqyCpws9EaAjKKAefuNP+z+8NY8khckgyHN2LL
 pfhv+iP8m+bF66HNDUlFz8ZrVOu3WQapgLPV90kIskNKXX3a

------END CERTIFICATE-----
```

Copy and save the text to a file, using a tool such as Notepad, giving it a name you can remember (for example, Certif.txt).

Use Setkey.exe (included in your Microsoft Internet Information Server installation) to install your signed certificate on the server, for example:

```
setkey MyPassword1 keypair.key certif.txt
```

**Note:** If you do not specify an IP address, the same certificate will be applied to all virtual servers created on the system. If you are hosting multiple sites on a single server, you can specify that the certificate only be used for a given IP address by adding the IP address as follows:

```
setkey MyPassword1 keypair.key certif.txt 10.191.28.45
```

### Configuring a Directory to Require SSL

Once the certificate has been applied, the SSL feature is enabled from Internet Service Manager for the WWW services. SSL can be required on any virtual directory available through the Internet server and is configured on the Directories pane of the Service property sheet.

### Suggestions for SSL Configuration and Operation

Microsoft recommends that you use separate content directories for secure and public content (for example, C:\Inetsrv\Wwwroot\Secure-Content and C:\Inetsrv\Wwwroot\Public-Content). It is important to avoid having a

server directory not protected by SSL as a parent for a secure directory.

It is suggested that you save your key file (Keypair.key) in a safe place in case you need it in the future. It is a good idea to store Keypair.key on a floppy disk and remove it from the local system after completing all setup steps. Do not forget the password you assigned to the key pair file.

**Setkey.exe Notes**

Do not specify a server name when running Setkey.exe on the local computer. After running Setkey.exe, restart the WWW service to enable SSL.

**Setting the Default Document and Directory Browsing**

If a remote user sends a request without a specific filename (for example, http://www.microsoft.com/), the WWW service will return the specified default document, if it exists in that directory. You can place a file with the specified default document filename in each subdirectory.

If no default document is available, the server will return an error, unless directory browsing is enabled. If directory browsing is enabled, a directory listing containing links to the files and directories in that directory will appear.

A default document can be included in all WWW directories. In the Directories property sheet for the WWW service, change the Default Document entry to the default file name you will use on your system. Often the default document is set to be an index file (Index.htm) for the contents of that directory (or of the entire server). The default file name used is Default.htm.

If the user does not specify a file for a particular directory, a hypertext file and directory listing will be returned.

Directory browsing on the WWW service is very similar to browsing in File Tranfer Protocol (FTP). Directory browsing is useful if you have a lot of files that you want to share quickly without converting them to HTML format.

**Note** Virtual directories will not appear in directory listings (also called directory browsing for the WWW service). To access a virtual directory users must know the virtual directory's alias, and type the URL address in their browser. For the WWW service, you can also create links in HTML pages. For the Gopher, service you can create explicit links in tag files for users to access virtual directories. For the FTP service, you can list virtual directories using directory annotations.

**Understanding Virtual Servers**

By convention each domain name, such as www.company.com, represents an individual computer. However, it is possible to use a single computer and make it appear to be not only a primary server (for example, named www.company.com), but also servers for different departments of your company (for example, marketing.company.com, sales.company.com, and so on). You can create "virtual servers" for these departments with Microsoft Internet Information Server. You do not need a different computer for each domain name.

To do this, you must obtain Internet Protocol (IP) addresses from your Internet Service Provider (ISP) for the primary server and for each virtual server you want to create. For example, you assign the first IP address (10.212.56.184) in the Domain Name System (DNS) as www.company.com (your primary server), and assign C:\Inetsrv\Wwwroot as its content home directory. You register the second IP address (10.212.56.185) in DNS as marketing.company.com, and assign a different drive or directory as its content home directory. Thus, it appears to users on the Internet that there are two computers when in fact it is the same computer running one copy of the WWW service. If you create a home directory without specifying an IP address, that home directory will be used for all requests containing server IP addresses not specified in other home directories.

These multiple IP addresses can be assigned to multiple network adapter cards, or to a single card. You use the Network applet in the Windows NT Control Panel to bind the additional IP addresses to your network adapter card.

After the IP address is assigned to the network adapter card, you must assign a home content directory to that IP address. In the Directories property sheet, select the Virtual Server box and enter its IP address. Virtual directories (directories that are not home directories) can also be restricted to one virtual server by assigning an IP address to them.

To create more than five virtual servers you must change a Windows NT Registry entry.

**Specifying Directories for Virtual Servers**

If you have assigned more than one IP address to your server, when you create a directory you must specify which IP address has access to that directory. If no IP address is specified, that directory will be accessible to all virtual servers.

**Important:** The default directories created during setup do not specify an IP address. You may need to specify IP addresses for the default directories when you add virtual servers.

To specify the fully qualified path for the directory to use for the selected virtual server, choose the Add button and type the path in the Directory box of the Directory Properties dialog box, or use the Browse button to pick the directory to use.

**Configuring Content Directories**

If your site is complex, you can configure Internet Information Server to publish from multiple directories in your service by using Internet Service Manager. The Directories property sheet lists the content directories used by the WWW service.

Click the Add button or the Edit Properties button in the Directories property sheet to configure individual WWW service directories.

**Home Directory**

Every server must have a home directory for content files. The home directory is the "root" directory for that service. As a root directory it does not have a name. By default, the home directory and all subdirectories are available to users.

**Virtual Directory**

Choose this to specify additional directories that are addressable as subdirectories of the WWW service root directory.

For example, if you want to provide three different product catalogs, each catalog could be stored on a separate hard drive on the server www.company.com.

To browsers, virtual directories are addressable as subdirectories off the "root" home directory. You must provide the name (alias) that browsers will use to specify that directory.

You can create an almost unlimited number of virtual directories for your service, although performance may suffer if you create too many of them. Use them only as required.

**Note** Virtual directories will not appear in directory listings (also called directory browsing for the WWW service). To access a virtual directory users must know the virtual directory's alias, and type the URL address in their browser. For the WWW service, you can also create links in HTML pages. For the Gopher, service you can create explicit links in tag files for users to access virtual directories. For the FTP service, you can list virtual directories using directory annotations.

**Specifying Directories with Virtual Servers**

If you have assigned more than one IP address to your server, when you create a directory you must specify which IP address has access to that directory. If no IP address is specified, that directory will be visible to all virtual servers.

**Important:** The default directories created during setup do not specify an IP address. You may need to specify IP addresses for the default directories when you add virtual servers.

**Account Information**

This entry applies only if the physical directory is listed by using a Universal Naming Convention (UNC) path, such as \\Research4\Public\WWWfiles. Enter a username and password with permission to use the network directory share. Virtual directories on network drives must be on computers in the same Windows NT domain as the Internet Information Server.

**Access Check Boxes**

**Read** must be selected for content directories.

**Write** must be selected for directories that will accept data from users. Assign Write access cautiously to prevent unauthorized users from placing destructive files on your computer.

**Execute** must be selected for directories containing programs, scripts, and Internet Server API (ISAPI) applications. Ensure that any directory marked Execute is not also marked Write; (this will prevent destructive programs from being copied to your server and then started, causing damage to your system. Also, ensure that any directory marked Execute is not also marked Read; this will prevent users from seeing your interactive content files.

**Require secure SSL channel** must be selected to require encrypted communication for a directory. For more information on Secure Sockets Layer (SSL), see SSL.